ISB Cancer Genomics Cloud Documentation

Release 1.0.0

the ISB-CGC team

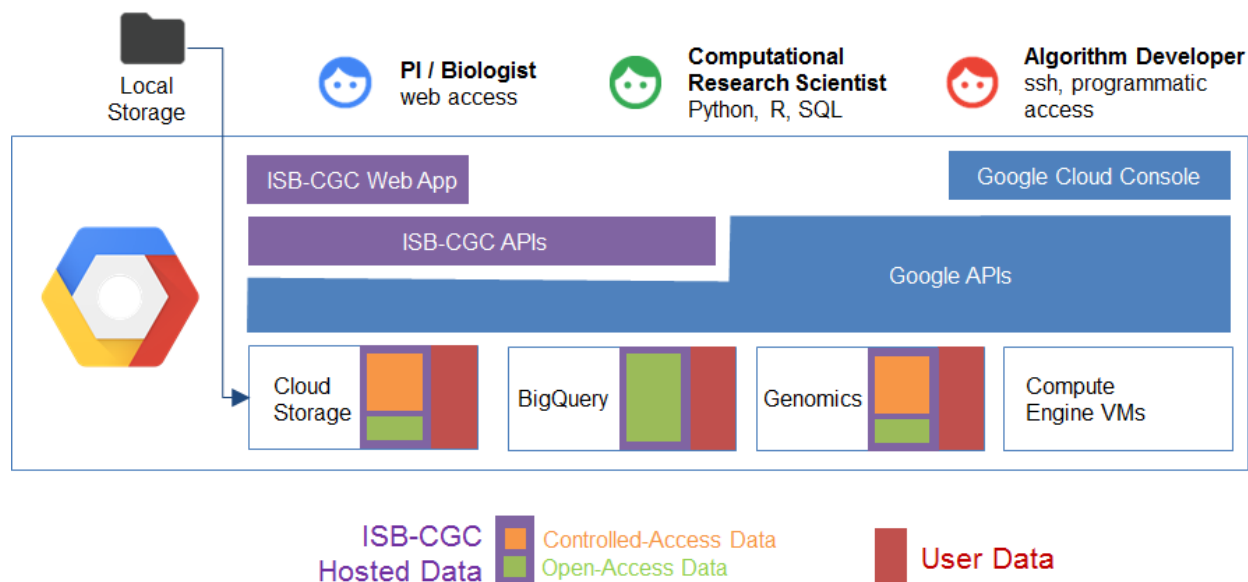**Jul 03, 2019**

Contents

Welcome to the ISB-CGC Documentation on Read the Docs.

Here you will find information describing the features of the ISB-CGC platform, tips on how to use it, and details about the data that we are hosting on the Google Cloud Platform.

The ISB-CGC aims to serve the needs of a broad range of cancer researchers ranging from scientists or clinicians who prefer to use an interactive web-based application to access and explore the rich TCGA, TARGET, CCLE, and COSMIC datasets, to computational scientists who want to write their own custom scripts using languages such as R or Python, accessing the data through APIs, and to algorithm developers who wish to spin up thousands of virtual machines to analyze hundreds of terabytes of sequence data.

This documentation is a work-in-progress, please let us know how we can improve it. feedback@isb-cgc.org

– the ISB-CGC team

Contents

1.1 About the ISB-CGC

The ISB-CGC provides interactive and programmatic access to all data hosted by institutes such as the Genomic Data Commons GDC and the Wellcome Trust Sanger Institute, leveraging many aspects of the Google Cloud Platform including BigQuery, Compute Engine, App Engine, Cloud Datalab and Google Genomics. Open-access clinical and biospecimen information for all TCGA, TARGET, CCLE, and COSMIC patients and samples, combined with the analyzed summary data and genomic reference and platform-annotation sources are stored in BigQuery, enabling fast SQL-like queries against the entire dataset. Controlled-access DNA and RNA sequence data is available to dbGaP-authorized users in the original BAM and FASTQ file formats for TCGA and TARGET data at this time.

The ISB-CGC aims to serve the needs of a broad range of cancer researchers ranging from scientists or clinicians who prefer to use an interactive web-based application to access and explore the rich TCGA, TARGET, CCLE, and COSMIC datasets, to computational scientists who want to write their own custom scripts using languages such as R or Python, accessing the data through APIs, to algorithm developers who want to spin up thousands of virtual machines to rapidly analyze hundreds of terabytes of sequence data. The ISB-CGC allows scientists to interactively define and compare cohorts, examine the underlying molecular data for specific genes or pathways of interest, and share insights with collaborators around the globe.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.2 How To Get Started on ISB-CGC

The ISB-CGC provides both interactive (through a web application) and programmatic access to data hosted by institutes such as the Genomic Data Commons GDC of the National Cancer Institute (NCI), and the Wellcome Trust Sanger Institute, leveraging many aspects of the Google Cloud Platform.

If you would like this document in pdf format please click here.

More about ISB-CGC, ISB-CGC Main Landing Page, Full documentation of the ISB-CGC platform, and FAQS.

1.2.1 Benefits of Using The Cloud

You don't have to download the data! Bring your compute and know-how to the data. Use cloud-native, compute scale as big as you can imagine, tools to analyze TBs and PBs of data!!

Most bioinformaticians today are most likely accustomed to using the high performance compute (HPC) resources provided by their institution to conduct high-throughput bioinformatics analyses. Here's a breakdown on how the Google Cloud Platform compares to your institution's HPC resources.

	Your University's HPC Resource	Google Cloud Platform
Operating Systems	Linux, Windows	Virtual machines can run Linux and Windows
Compute	Virtual machines not determined by you	You can sign up with you own virtual machines*
Storage	Block Storage • Small storage is avaialable in your home directory (usually around 1TB) • Some Scratch storage that is often deleted after a certain amount of time • Storage is usually a shared resource	Block Storage & Object Storage • Each virtual machine instance has a single boot persistent disk with a default size of 10GB that can be adjusted up to 64TB* • For storage that needs IO, consider persistent disks • Google Cloud Storage (GCS) buckets are the most flexible and economical storage option • You can save objects to GCS buckets including images, videos, blobs, and unstructured data
Pricing	Depends on the institution: • Instituion provides basic HPC resources for researchers free of charge • PIs requiring larger-scale resources must purchase clusters and storage space	Pay as you go • You pay for the compute resources and storage that you use*
Do you have to wait?	Yes • Resources are shared amongst users • Scheduler systems used to schedule jobs based on resource availability	No • Once you've set up a Google Cloud Platform account, you can spin up a virtual machine and begin computing quickly
Is machine powerful enough?	Yes and no, depends on what you're trying to do often no	Compute resources and storage are unlimited but, you have to pay for it*
Acessing Cancer Genomics Data	Typically not stored on the HPC, you have to download to your local machine	Data is stored on the cloud
How to connect	Log in using Secure Shell protocal (SSH)	Log in using Secure Shell protocal (SSH)

*Be careful of costs

1.2.2 Goals of this tutorial:

1. How to get started on the ISB-CGC powered by the Google Cloud Platform (GCP)

2. Learn about cloud credits offered by Google and ISB-CGC

3. Learn about data hosted on ISB-CGC

4. Learn about how to get authorization to access controlled cancer genomics data

5. Setting up and registering a GCP project to use controlled access data

6. Enabling required GCP APIs

7. Learn about GCP's BigQuery tool

8. Learn how to run analyses on data stored in BigQuery or stored in Google Cloud Storage (GCS)

1.2.3 Take home message:

Don't be intimidated by the cloud! Bring your computation to the data on ISB-CGC. If you've conducted bioinformatics analyses before using the command line or SQL, this will be just as easy (if not easier).

I. Getting Started:

1.) ISB-CGC hosts both open-access and controlled-access cancer genomics data from the NCI.
About ISB-CGC Cloud-Hosted Datasets

2.) To access controlled-access data, dbGaP authorization is required. Accessing Controlled-Access Data and acquiring dbGaP authorization

3.) To work in GCP, you must first set up a GCP Project:

- A GCP project is required to make use of all of the data, tools, and Google Cloud functionality.

- Do you have a Google identity already (e.g. a GMail account)? Your institutional email may be a Google identity (if your institution uses Google Apps), or you may have a personal GMail address.

- If not, it only takes a minute to create a google identity. You can even link a non-GMail account (eg. scientist@nih.gov) as a Google identity by this method.

- Create your own GCP project and take advantage of a one-time $300 Google Credit.

- If you have already used this one-time offer (or there is some other reason you cannot use it), please see the information here about ISB-CGC Cloud Credits Available for Researchers.

- How to request ISB-CGC Cloud Credits.

4.) Registering the GCP project

5.) Enable Required Google Cloud APIs

II. Accessing and analyzing data via BigQuery

- BigQuery is Google's native big data analysis tool. It is a serverless, highly scalable data warehouse tool that allows researchers to find meaningful insights from data using standard SQL queries CHEAPLY, and FAST!

- ISB-CGC has leveraged this powerful tool and uploaded multiple cancer genomics datasets into BigQuery tables that are open to the public. ISB-CGC Datasets in BigQuery and the always freshly updated Data Release Notes and Future Plans .

- To obtain access to the ISB-CGC open access project tables in BigQuery, users can link these tables to your GCP project as described here.

- To obtain access to the ISB-CGC controlled access project tables in BigQuery, users can link these tables to your GCP project as described here.

- ISB-CGC provides tutorials and walkthroughs on how to access BigQuery from the web-UI, programmatically in R, or through Google's native Jupyter notebook Cloud Datalab, and python examples.

- Every month, ISB-CGC provides an example analysis of cancer genomics data using BigQuery in our Query of the Month blog.

III. Accessing and analyzing data stored in GCS

- All open-access data on ISB-CGC are stored in a publically available GCS bucket (gs://isb-cgc-open).

- All controlled-access data are stored in Google Cloud Storage (GCS) in their original form as obtained from the GDC.

- To access controlled data, users must first be authenticated by NIH (via the ISB-CGC web-app). Upon successful authentication, user dbGaP authorization will be verified. These two steps are required before the user's Google identity is added to the access control list (ACL) for the controlled data. At this time, this access must be renewed every 24 hours.

- Summary of data types and format available

- New programs have been added into Google Cloud Storage, directions on how to find the data paths via IndexD can be found here.

- Working with large-scale data hosted by the ISB-CGC in Google Cloud Storage requires some familiarity with tools such as the Google Cloud SDK, Google Compute Engine, Virtual Machines and Docker.

- Cheat-sheets and slides on computing in the cloud including how to access files stored on GCS can be found here.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.3 Best Practices

1.3.1 DON'T DOWNLOAD THE DATA

The ISB-CGC platform is one of NCI's Cancer Genomics Cloud Resources and our mission is to host TCGA and TARGET data in the cloud so that researchers around the world may work with the data without needing to download and store the data at their own local institutions.

Remember those times when you had to wait weeks to download the data - you don't need to do that any more! The data is already on the cloud, so you can collaborate with other researchers much more easily. Be mindful that if you download data, you'll incur egress charges. Google egress charges information

1.3.2 COMPUTING ON THE CLOUD

Most of the same linux commands, scripts, pipelines/workflows, genomics software packages and docker containers that you run on your local machine can be executed on virtual machines on the Google cloud.

a.) The basics and best practices on how to launch virtual machines (VMs) are described here in our documentation. **NOTE: When launching VMs, please maintain the default firewall settings.**

b.) Compute Engine instances can run the public images for Linux and Windows Server that Google provides as well as private custom images that you can create or import from your existing systems.

Be careful as you spin up a machine, as the larger machines cost you more. If you are not using a machine, shut it down, you can always restart it easily when you need it.

Example use-case: You would like to run Windows-only genomics software package on the TCGA data. You can create a Windows based VM instance.

c.) More details on how to deploy docker containers on VMs are described here in Google's documentation: deploying containers

d.) A good way to estimate costs for running a workflow/pipeline on large datasets is to test them first on a small subset of data.

e.) There are different VM types depending on the sort of jobs you wish to execute. By default, when you create an VM instance, it remains active until you either stop it or delete it. The costs associated with VM instances are detailed here: compute pricing

f.) If you plan on running many short compute-intensive jobs (for example indexing and sorting thousands of large bam files), you can execute your jobs on preemptible virtual machines. They are 80% cheaper than regular instances. preemptible vms

Example use-cases:

- Using preemptible VMs, researchers were able to quantify transcript levels on over 11K TGCA RNAseq samples for a total cost of $1,065.49.

  Tatlow PJ, Piccolo SR. A cloud-based workflow to quantify transcript-expression levels in public cancer compendia. Scientific Reports 6, 39259

- Also Broad's popular variant caller pipeline, GATK was designed to be able to run on preemptible VMs.

1.3.3 STORAGE ON THE CLOUD

The Google Cloud Platform offers a number of different storage options for your virtual machine instances: disks

a.) Block Storage:

- By default, each virtual machine instance has a single boot persistent disk that contains the operating system. The default size is 10GB but can be adjusted up to 64TB in size (Be careful! High costs here, spend wisely!).

- Persistent disks are restricted to the zone where your instance is located.

- Use persistent disks if you are running analyses that require low latency and high-throughput.

b.) Object Store: Google Cloud storage buckets are most the flexible and economical storage option.

- Unlike persistent disks, Cloud Storage buckets are not restricted to the zone where your instance is located.

- Additionally, you can read and write data to a bucket from multiple instances simultaneously.

- You can mount a GCS bucket to your VM instance when latency is not a priority or when you need to share data easily between multiple instances or zones.

An example use-case: You want to slice thousands of bam files and save the resulting slices to share with a collaborator who has instances in another zone to use for downstream statistical analyses. - You can save objects to GCS buckets including images, videos, blobs and unstructured data.

A comparison table detailing the current pricing of Google's storage options can be found here: storage features

---

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.4 Cloud-Hosted Data Sets

The ISB-CGC platform hosts the majority of the TCGA data set as well as other reference and annotation datasets in different appropriate Google Cloud technologies:

- low-level DNA- and RNA-Seq data are stored primarily in Google Cloud Storage;
- high-level clinical, biospecimen, and molecular data are available in a series of carefully curated datasets and tables backed by the massively-parallel analytics engine Google BigQuery;
- TCGA radiology and tissue image data are now also available in Google Cloud Storage;
- TCGA proteomics (CPTAC PhaseII) data has also been uploaded to Google Cloud Storage;

The original mission of the ISB-CGC was to host the TCGA dataset. We are now in midst of adding data from the TARGET pediatric cancer. Stay tuned for updates.

1.4.1 A note about legacy and harmonized data sets

Programs like TCGA that pre-date the Genomic Data Commons will have both legacy data sets (data as originally generated by the program) and harmonized data sets created by the Genomic Data Commons. Whiles these data sets do have much in common, as the GDC works through its harmonization process a number of changes can occur including removal or addition of cases and samples or changes in terminology. One of the goals of the ISB-CGC is to stay current with changes introduced by GDC and therefore you may find differences between legacy data and harmonized data.

NCI Cancer Programs

In recent years, the National Cancer Institute, in collaboration with other institutes within NIH, has invested in the production and analysis of several large datasets. The ISB-CGC platform is funded by NCI in an effort to make these data more accessible and usable.

The initial goal of the ISB-CGC was to host the data produced by The Cancer Genome Atlas program. We are now expanding to host data from TARGET (a pediatric cancer program), and will in the future host data from newer projects supported by the Cancer Genome Characterization Initiative.

Please see the individual sections below for more information about these individual, large-scale programs.

TCGA Overview

The Cancer Genome Atlas (TCGA) is a comprehensive and coordinated effort to accelerate the understanding of the molecular basis of cancer through the application of genome analysis technologies, including large-scale genome sequencing.

The overarching goal of TCGA is to improve our ability to diagnose, treat and prevent cancer. To achieve this goal in a scientifically rigorous manner, the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) used a phased-in strategy to launch TCGA. A pilot project developed and tested the research framework needed to systematically explore the entire spectrum of genomic changes involved in more than 20 types of human cancer.

This massive effort was launched in 2006. The final samples were shipped in mid-2014, and analysis of the data produced by this program continues to this day.

For more information please visit the official TCGA website.

About the TCGA Data

The ISB-CGC hosts approximately 1 petabyte of TCGA data in Google Cloud Storage (GCS) and in BigQuery.

The ISB-CGC platform is one of NCI's Cancer Genomics Cloud Pilots and our mission is to host the TCGA data in the cloud so that researchers around the world may work with the data without needing to download and store the data at their own local institutions.

The vast majority (over 99%) of this **petabyte** of data consists of low-level sequence data, currently stored as files in Google Cloud Storage (see figure below). Over the course of the TCGA project, this low-level (*"Level 1"*) data has been processed through a set of standardized pipelines and the the resulting high-level (*"Level 3"*) data is frequently the data that is used in most downstream analyses. The ISB-CGC platform aims to make these different types of data accessible to the widest possible variety of users within the cancer research community, using the most appropriate Google Cloud Platform technologies.
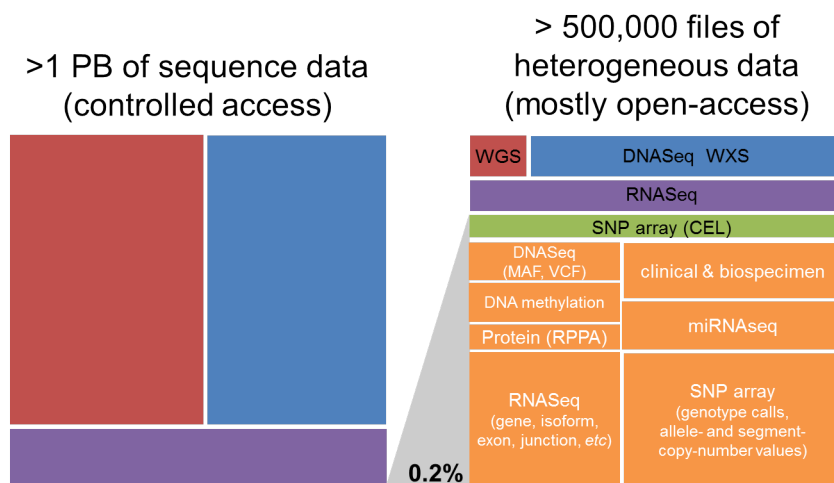
More details about the TCGA data-generating platforms, data-types, and levels and can be found in the sections below:

Understanding the TCGA Data Platforms

When working with any of the data types, it is important to also be aware of both the *platform* that was used to generate the underlying raw data as well as the *pipeline* that was used to process the data. For example, over the course of the

TCGA study, DNA methlyation data was obtained using first the Illumina HumanMethylation27 platform, and later using the HumanMethylation450 platform. Any analysis that combines data from these two platforms across a cohort of samples should take this into consideration. Another example where multiple platforms and/or pipelines were used to produce a single data type is the Level-3 gene expression data: most tumor samples were processed at UNC and the normalized gene-expression values are based on the RSEM method, while some tumor samples were processed at BCGSC and the normalized gene-expression values are based on RPKM.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Understanding the TCGA Data Types

The TCGA dataset is unique in that the tumor samples were assayed using a standard set of platforms and pipelines in order to produce a comprehensive dataset including:

- DNA sequencing of tumor samples and matched-normals (typically blood samples) in order to detect somatic mutations;
- SNP array based DNA copy-number and genotyping analysis of tumor samples and matched-normals;
- DNA methylation of tumor samples;
- messenger RNA (mRNA) expression analysis of the tumor samples to capture the gene expression profile;
- micro-RNA (miRNA) expression profiling of the tumor samples;

In addition, protein expression for a significant fraction (~20%) of all tumor samples was obtained using RPPA (reverse phase protein array).

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Understanding the TCGA Data Levels

TCGA Data Levels

For each *type* of data, there are typically three *levels* of data: * Level 1 typically represents raw, un-normalized data * Level 2 typically represents an intermediate level of processing and/or normalization of the data; * Level 3 typically represents aggregated, normalized, and/or segmented data.

The results of integrative or pan-cancer analyses are sometimes referred to as "Level 4" data. More information about Data Level Classification can be found on the NCI wiki.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

In addition, we recommend that you review important information about data security and data access in these sections:

Understanding Data Security

Much of the low-level TCGA and TARGET data (including DNA and RNA reads, and SNP CEL files, for example) are classified as "controlled access data" and are under the control of the dbGaP Data Access Committee (DAC).

Investigator(s) requesting to receive Genomic data in accordance with the NIH Genomic Data Sharing Policy are required to submit:

- a **data access request** (DAR)

- a **research use statement** (RUS)

Note: Requesters and institutional signing officials (SO) must have NIH eRA user IDs to begin this process. Visit the electronic Research Administration (eRA) for more information on registering for a NIH eRA account. NIH staff may utilize their NIH log-in. (See the dbGaP Data Access Request Portal for additional instructions.)

Additionally, they must:

- Submit a Data Use Certification (DUC) co-signed by the designated Institutional Official(s) at their sponsoring institution (sample DUC);

- Protect data confidentiality (any data which has been designated "controlled" **must** be protected accordingly, unless prior release authorization is obtained from a NCI data custodian); and

- Ensure that appropriate data security measures are in place.

In the context of Google Cloud Platform (GCP) projects, it is important to realize that all members of a GCP project have (at least) read access to all data stored within that project, as well as to all virtual machines, boot-disks, and persistent disks attached to that project. Therefore, if a PI establishes a GCP project (project-A) for the purposes of analyzing controlled data (*eg* performing mutation analysis on TCGA sequence data), then *all* members of project-A must be authorized to view controlled data. The outputs of certain analyses performed on controlled data, if they are summary in nature, may no longer be controlled data and could be copied to a second GCP project (project-B) for further downstream analyses by researchers who are not authorized to view controlled data. Researchers who are not authorized to view controlled data could be made members of project-B, while users who *are* authorized could be members of both project-A *and* project-B.

Note: The PI and the PI's institution are *responsible* for and will be held *accountable* for ensuring the security of controlled data, not the cloud service provider. The Google Cloud Platform has been certified as FedRAMP compliant which means that it has been independently assessed and shown to meet all necessary FedRAMP security controls. This provides the assurance that the data-security and access control mechanisms implemented by the Google Cloud Platform and made available to end-users are sufficient to safeguard the data. However, it remains the PI's responsibility to ensure that these access control mechanisms are used appropriately and effectively within the context of the PI's GCP project.

You should think about securing controlled data within the context of your GCP project in the same way that you would think about securing controlled data that you might download to a file-server or compute-cluster at your own institution. Your responsibilities regarding the appropriate use of the data are the same in a cloud environment. For more information, please refer to NIH Security Best Practices for Controlled-Access Data.

"The Investigator and their associated institution assume the responsibility for the security of the dbGaP data. As such, NIH has tried to provide as much information as possible for PIs, institutional signing officials (SOs) and the IT staff who will be supporting these projects, to make sure they understand their responsibilities." (Ref: The Cloud, dbGaP and the NIH blog post 03.27.2015)

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Understanding Data Access

- **Public Data** Sometimes the word "public" is misinterpreted as meaning "open". All of the TCGA data is *public* data, and much of it is *open*, meaning that it is accessible and available to *all* users; while some low-level TCGA data is *controlled* and restricted to authorized users.

- **Open-Access Data** Depending on how you categorize the data, *most* of the TCGA data is open-access data. This includes all de-identified clinical and biospecimen data, as well as all Level-3 molecular data including gene expression data, DNA methylation data, DNA copy-number data, protein expression data, somatic mutation calls, etc.

- **Controlled-Access Data** All low-level sequence data (both DNA-seq and RNA-seq), the raw SNP array data (CEL files), germline mutation calls, and a small amount of other data are treated as *controlled* data and require that a user be properly authenticated and have dbGaP-authorization prior to accessing these data.

Note that many public, open-access datasets may still be **restricted** in various ways. Typically, a **License** document containing explicit terms of use will be associated with each dataset. Some institutions have their own licenses, though many use one of the Creative Commons licenses. License terms apply to both data and source-code, so please be aware of the terms of a license whenever you plan to re-use data or source-code produced by someone else.

In the earlier days of the TCGA data, although the data was made public as quickly as possible, it was generally under **embargo** for some period of time, to allow the TCGA analysis working groups to produce the initial "marker paper" for each tumor type. Now that the TCGA project is nearing completion, none of the TCGA data is under embargo anymore, but we still recommend that you review the TCGA Publication Guidelines.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Historically, the data being hosted by the ISB-CGC was obtained from two former TCGA data repositories:

- **TCGA DCC**: the TCGA Data Coordinating Center which provided a Data Portal from which users could download open-access or controlled-access data. This portal provided access to all TCGA data *except* for the low-level sequence data.

- **CGHub**: the Cancer Genomics Hub was NCI's current secure data repository for all TCGA BAM and FASTQ sequence data files.

As of June 2016, the official data repository for all TCGA and other NCI CCG data is the NCI Genomic Data Commons. The original TCGA data, aligned to the hg19 human reference genome is available from the NCI-GDC's legacy archive while the new "harmonized" data, realigned to hg38 is available from the NCI-GDC's main data portal.

For more information about the original data source repository and data access classes (open *vs* controlled), please refer to these sections:

TCGA Data by Access Class

Open-Access TCGA Data

The open-access TCGA data hosted by the ISB-CGC Platform includes:

- Clinical (de-identified) and Biospecimen data: these data were originally provided in XML files (Level-1) by the DCC;

- Somatic mutation data: these data were originally provided in MAF files (Level-2) by the DCC;

- DNA copy-number segments: these data were originally provided as segmentation files (Level-3) by the DCC;

- DNA methylation data: these data were originally provided as TSV files (Level-3) by the DCC;

- Gene (mRNA) expression data: these data were originally provided as TSV files (Level-3) by the DCC;

- microRNA expression data: these data were originally provided as TSV files (Level-3) by the DCC;

- Protein expression data: these data were origially provided as TSV files (Level-3) by the DCC; and

- TCGA Annotations data: annotations were originally obtained from the TCGA Annotations Manager, and can now also be found on the

GDC Data Portal

in Google Cloud Storage (GCS)

The data files described above are available to all ISB-CGC users in an open-access GCS bucket (gs://isb-tcga-phs000178-open).

in BigQuery

The information scattered over tens of thousands of XML and TSV files at the DCC is provided in a *much more accessible* form in a series of BigQuery tables.

For more details, including tutorials and code examples in Python or R, please see our github repositories.

This introductory tutorial gives a great overview of all of the tables and pointers on how to get started exploring them. Be sure to check it out!

Controlled-Access TCGA Data

The controlled-access TCGA data hosted by the ISB-CGC Platform includes:

- SNP array CEL files: these Level-1 data files were provided by the DCC and include over 22,000 files for both tumor and matched-normal samples;
- VCF files: these Level-2 data files were provided by the DCC and include over 15,000 files produced by several different centers (primarily Broad and BCGSC);
- MAF files: these "protected" mutation files (Level-2) were provided by the DCC (note that these files were not generated uniformly for all tumor types);
- **DNA-seq BAM files: these Level-1 data files were provided by CGHub;**
    - over 37,000 of these files are available in Google Cloud Storage (GCS);
    - roughly 90% of these BAM files contain exome data, the remaining 10% contain whole-genome data;
    - BAM index (BAI) files are also available for all BAM files;
- **mRNA- and microRNA-seq BAM files: these Level-1 data files were provided by CGHub;**
    - over 13,000 mRNA-seq BAM files are available in GCS;
    - over 16,000 miRNA-seq BAM files are available in GCS;
- mRNA-seq FASTQ files: these Level-1 data files were provided by CGHub and include over 11,000 tar files.

in Google Cloud Storage

At this time, all of these controlled-access data files are stored in GCS in the original form, as obtained from the data repository.

In order to access these controlled data, a user of the ISB-CGC must first be authenticated by NIH (via the ISB-CGC web-app). Upon successful authentication, the users's dbGaP authorization will be verified. These two steps are required before the user's Google identity is added to the access control list (ACL) for the controlled data. At this time, this access must be renewed every 24 hours.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

TCGA Data by Source Repository

TCGA Data at the NCI-GDC

In July 2016, both the DCC and CGHub were shut down. The official repository for all NCI datasets, including the TCGA data is now the NCI Genomic Data Commons.

Some information about the TCGA data at the DCC and CGHub is preserved below for historical reasons, and in case of need.

TCGA Data at the DCC

Complete sets of open-access and controlled-access data archives were copied from the DCC on October 4th, 2015 into Google Cloud Storage.

Note that for every archive at the DCC, there may be multiple revisions of an archive. A list of the current latest archives can be obtained from the DCC. The archive naming convention includes the disease code, the platform/pipeline name, the archive type (*eg* data level), the serial index (which is often the batch number), and the revision number. If you want to check whether there is a newer version of a specific archive at the DCC than what we currently have on the ISB-CGC platform, you can check the date column in the latest archive report mentioned above, or you can compare the archive name to these lists of open-access archives and controlled-access archives based on our most recent upload.

Note that all "bio" archives (containing clinical, biospecimen, and other types of XML files) were recently migrated to a new XSD which is not backwards compatible with the previous XSD. This update took place over the course of the month of December 2015 and none of these new archives are currently included in any of the current ISB-CGC BigQuery tables or files in GCS.

TCGA Data at CGHub

The complete listing of the (over 87,000) TCGA data files from CGHub that are currently available in Google Cloud Storage (GCS) contains the following four columns of information:

- unique CGHub id for the file,
- the TCGA aliquot barcode,
- the GCS object path, and
- the size of the file in bytes.

The final complete CGHub manifest (downloaded in early July, just before CGHub shut down) is also available

---

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

TARGET Overview

TARGET: Therapeutically Applicable Research to Generate Effective Treatments

TARGET applies a comprehensive genomic approach to determine molecular changes that drive childhood cancers. Investigators form a collaborative network to facilitate discovery of molecular targets and translate those findings into the clinic. TARGET is managed by NCI's Office of Cancer Genomics and Cancer Therapy Evaluation Program.

The TARGET data is available at the GDC, both in the legacy archive which contains over 10,000 files for over 5,000 cases. Virtually all of this data is low-level (and controlled-access) sequence data (including 1702 RNA-seq files, 765 miRNA-seq, with the remainder being WXS or WGS DNA-seq BAMs). Some of this data has been re-processed and is available on the main GDC Data Portal. This newer dataset so far includes 6183 files representing 3236 cases, and totaling over 17 TB. Over half of the files (3740) are controlled-access files, including BAM, VCF, and MAF file types, based on WXS, RNA-seq, and miRNA-seq data. The remaining 2443 open-access files, include RNA-seq and miRNA-seq quantification, as well as clinical and biospecimen supplement files.

For more information about the TARGET program, please visit the official TARGET website.

About the TARGET Data

The ISB-CGC currently hosts over 70 terabytes of TARGET data in Google Cloud Storage (GCS) and in BigQuery.

The ISB-CGC platform is one of NCI's Cancer Genomics Cloud Pilots and part of our mission is to host the TARGET data in the cloud so that researchers around the world may work with the data without needing to download and store the data at their own local institutions.

TARGET controlled-access data is now available to authorized users in Google Cloud Storage.

The open-access data includes RNA-seq and miRNA-seq expression levels, and is available in BigQuery, along with the open-access clinical and biospecimen information.

TARGET Data by Access Class

Open-Access TARGET Data

The open-access TARGET data hosted by the ISB-CGC Platform includes:

- Clinical (de-identified) and Biospecimen data: these data were originally provided in XML files (Level-1) by the TARGET DCC;
- Gene (mRNA) expression data: these data were originally provided as TSV files (Level-3) by the TARGET DCC;
- microRNA expression data: these data were originally provided as TSV files (Level-3) by the TARGET DCC;

in BigQuery

The information scattered over thousands of XLSX and TSV files at the GDC is provided in a *much more accessible* form in a series of BigQuery tables.

Controlled-Access TARGET Data

The controlled-access TARGET data is not yet accessible but will be soon, please stay-tuned for updates and let us know if you have an urgent need for this data. (Please note that you will need to obtain dbGaP authorization first, so if you do not yet have that, you should begin that process.)

in Google Cloud Storage

All controlled-access TARGET data will be available as files in GCS, in their original form (*ie* BAM or FASTQ files).

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

CCLE Overview

The Cancer Cell Line Encyclopedia (CCLE) project is an effort to conduct a detailed genetic characterization of a large panel of human cancer cell lines. The CCLE provides public access analysis and visualization of DNA copy number, mRNA expression, mutation data and more, for 1000 cancer cell lines.

The CCLE aligned reads (BAM files) are currently available in an open-access Cloud Storage bucket which you can browse here.

An older set of BigQuery tables containing CCLE data are available in the `isb-cgc:ccle_201602_alpha` dataset. This data will be updated and reformatted to look more like the newer TCGA and TARGET datasets after the GDC formally releases CCLE, to optimize usage in the cancer research community.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

CGCI Overview

CGCI: Cancer Genome Characterization Initiative

CGCI supports research to comprehensively catalog the genomic alterations in adult and pediatric cancers. The research community can use CGCI data to gain insight into the underlying mechanisms of these cancers and identify potential therapeutic targets.

For more information, please visit the official CGCI website.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

NCI-GDC Overview

The NCI's Genomic Data Commons (NCI-GDC) provides the cancer research community with a unified data repository that enables data sharing across cancer genomic studies in support of precision medicine.

The NCI-GDC Data Portal allows users to search for and download data directly via your web browser or using the NCI-GDC Data Transfer Tool. So-called "legacy" data that the NCI-GDC "inherited" from previous data coordinating centers (*eg* the TCGA-DCC and CGHub), is available in the Legacy Archive, while a "harmonized" data set (re-aligned to GRCh38/hg38 and re-processed by the NCI-GDC) is available at the main Data Portal. (We will generally refer to the harmonized/default archive available from the main NCI-GDC Data Portal as the "current" archive.)

The ISB-CGC is hosting much of this data (both "legacy" and "harmonized" in Google Cloud Storage (GCS), meaning that you may *not* need to download any data from the NCI-GDC if you're planning on running your analyses on the Google Cloud Platform. These tables can be previewed and queried conveniently and interactively from the BigQuery web UI or from scripting languages such as **R** and **Python**, or from the command-line using the cloud SDK utility **bq**. If you have used the NCI-GDC portal to create cohorts or filelists, you can follow these tutorials to bring that information into ISB-CGC for use.

In order to help users determine which data at the NCI-GDC is available on the ISB-CGC platform, we have created a set of metadata tables in BigQuery (based on NCI-GDC Data Release 8.0) in the isb-cgc:GDC_metadata dataset:

- rel8_caseData: contains a complete list of all 17268 cases existing in either the legacy or current archives. The following query, for example will return a count of the number of cases by program, together with the number of data files for those cases in the two archives:

```
SELECT
  project__program__name AS program,
  COUNT(*) AS numCases,
  SUM(legacy_file_count) AS totLegacyFiles,
  SUM(current_file_count) AS totCurrentFiles
FROM
  `isb-cgc.GDC_metadata.rel8_caseData`
GROUP BY
  project__program__name
ORDER BY
  numCases DESC
```

program	numCases	totLegacyFiles	totCurrentFiles
TCGA	11315	4000803	351699
TARGET	5003	19042	12491
CCLE	950	1273	0

(Note that some files contain data from *multiple* cases, and these types of files will be counted multiple times in the above query on this case-oriented table, resulting in an over-count of the number of unique files.)

- rel8_fileData_current: contains a complete list of the 274724 files in the current archive (268541 TCGA files and 6183 TARGET files)

```
SELECT
  cases__project__program__name AS program_name,
  experimental_strategy,
  data_category,
  data_format,
  data_type,
  COUNT(*) AS numFiles,
```

(continues on next page)

```
  SUM(file_size)/1000000000 AS totFileSize_GB
FROM
  `isb-cgc.GDC_metadata.rel8_fileData_current`
GROUP BY
  1, 2, 3, 4, 5
ORDER BY
  totFileSize_GB DESC
```

results of this query can be viewed here. The top three rows in the result are the TCGA WXS, TCGA RNA-Seq, and TARGET WXS BAM files, which total approx 350 TB, 100 TB, and 10 TB respectively.

- rel8_fileData_legacy: contains a complete list of the 805907 files in the legacy archive (718064 TCGA files, 10154 TARGET files, 1273 CCLE files, and 76416 files which are not currently linked to any program or project – 15386 of these are controlled-access files with the TCGA dbGaP identifier, and the remaining 61030 open-access files include ~17k coverage WIG files, ~12k diagnostic SVS images, ~11k clinical/biospecimen xml files). The results of the same query as above (but directed at this table) can be viewed here. The top two rows in the result are the TARGET and TCGA WGS BAM files, totaling over 600 TB and 500 TB respectively.

- rel8_aliquot2caseIDmap: is a "helper" table in case you need to be able to map between identifiers at different levels. A total of 164911 unique aliquots are identified in this table. The intrinsic hierarchy is program > project > case > sample > portion > analyte > aliquot. We use the term "barcode" where the NCI-GDC uses the term "submitter id", "gdc_id" for the NCI-GDC's uuid-style identifier. If a portion was not further divided into analytes or if an analyte was not further divided into aliquots, some of the fields in this table may simply have the string "NA". For example, this query for a single TCGA case will return 24 rows of results for 2 unique samples, 1 portion from each sample, 5 analytes from the tumor sample and 3 analytes from the blood-normal sample, and finally 24 unique aliquots total.

```
SELECT
  *
FROM
  `isb-cgc.GDC_metadata.rel8_aliquot2caseIDmap`
WHERE
  case_barcode="TCGA-23-1029"
ORDER BY
  aliquot_barcode
```

- rel8_slide2caseIDmap: is another very similar "helper" table, but for the tissue slide data. A total of 18682 slide identifers are included. In this table the hierarchy is program > project > case > sample > portion > slide.

- rel8_GDCfileID_to_GCSurl: is the table to use to determine whether and where a particular NCI-GDC file is available in Google Cloud Storage (GCS). Between the two NCI-GDC archives (legacy and current), there are over one million files. Of these, over 500000 files, totaling over 1700 TB, are available in ISB-CGC buckets in GCS. This SQL query, for example, can be used to get summaries of the NCI-GDC data that is available in GCS (sorted according to the total size in TB):

dbName	access	program_name	n	total_size_TB
legacy	controlled	TCGA	177490	1132.209488904317
current	controlled	TCGA	44402	451.901815431968
legacy	controlled	TARGET	6763	62.997475267139
legacy	open	CCLE	1273	22.805178386465
current	controlled	TARGET	1459	15.565503418139
legacy	open	null	22874	12.976994070637
legacy	open	TCGA	230319	4.725815391355
current	open	TCGA	22475	0.00126935315
legacy	controlled	null	19	3.50647065E-4

Let's take a closer look (SQL) at the large number of open-access files that are *not* available in GCS, looking specifically at files where the `data_format` is either `TXT` or `TSV` and see what types of data that represents. The complete results of this query can be found here.

Much of this type of data is provided by ISB-CGC in BigQuery tables rather than the raw flat files, where the data is more easily explored using Standard SQL backed a massively-parallel analytics engine and also accessible from R or Python. Fore more details, please see our Data in BigQuery section.

Conversely, if we take a look at data that is *not* available in GCS, and is *not* of the `TXT` or `TSV` type which would be amenable to putting into BigQuery tables. We find that the single largest category of data at the NCI-GDC which is not currently available in any ISB-CGC buckets consists "raw" Methylation array data, DNA-Seq coverage (WIG) files, "raw" Protein expression array data, clinical pathology reports, etc.

Please let us know if there are any important data sets at the GDC that you would like to see made available in ISB-CGC cloud buckets.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

ISB-CGC Hosted Data Sets

Part of the mission of the ISB-CGC has been to explore the best ways to use the available cloud technologies to provide access to the hosted data. To this end, the hosted data is made available using these three main Google Cloud Platform technologies:

- **Google BigQuery (BQ),** a massively-parallel analytics engine is ideal for working with data that is essentially tabular in nature. This includes, the high-level clinical, biospecimen, and molecular data from the main NCI programs. It is also where we store a large amount of metadata about files that are more appropriately stored in Google Cloud Storage, as well as genome reference sources (*eg* GENCODE, miRBase, *etc*). All of these datasets and tables are completely *open access* and available to the research community.

- **Google Cloud Storage (GCS),** a cloud-hosted object-store is used to store other types of (typically binary) data which is typically processed by custom software pipelines. In our case this means the low-level sequence data, in BAM or FASTQ format, as well as pathology and radiology images (in SVS or DICOM format). All controlled-access data is currently only available in GCS – access to these data requires that a user walk through the required authentication and authorization steps.

- **Google Genomics (GG),** provides a new way to work with sequence-level data, via the GA4GH API. If and when the research community shifts away from BAM files towards using the GA4GH API, using this technology as our primary data-store may make more sense.

Please refer to the sections below for more details about the data available in these three Google Cloud technologies:

Data in BigQuery

The information scattered over tens of thousands of XML and TSV files in two separate archives at the NCI-GDC is provided in a *much more accessible* form in a series of *open-access* BigQuery tables. For more details, including tutorials, SQL, and code examples in Python or R, please see our Query of the Month page and our github repositories. Note that dbGaP authorization is **not** required to access these tables!

If you have suggestions or requests for additional data (*eg* TCGA isoform expression data, or other reference data sources) that you would like to see made available as BigQuery tables, please let us know (feedback@isb-cgc.org) and we will try to make that happen.

BigQuery Datasets and Tables

Data made available by the ISB-CGC through BigQuery is organized into several *open-access* datasets, where a dataset is made up of multiple tables. Datasets in BigQuery are uniquely identified based on the Google Cloud Platform (GCP) project name (in this case **isb-cgc**), and the dataset name, separated by a colon (or a period, in standard SQL), *eg* `isb-cgc:TCGA_bioclin_v0`. Tables are uniquely identified by appending the table name, preceded by a period, *eg* `isb-cgc:TCGA_bioclin_v0.Clinical`.

You can use the Google BigQuery graphical interface to query the BigQuery datasets. To view and use these datasets, please see this page for accessing controlled data in BigQuery or this page to view open access data in BigQuery.

The following sections describe each of the major datasets that are currently publicly-accessible, and the tables that each one contains. For additional details regarding the ETL (extract, transform, and load) process for each of these data types, please refer to the data-type specific details below.

For a more visual overview of the contents of BigQuery and how they relate to one another, you might find this view useful.

TCGA Clinical, Biospecimen and Molecular Data

The TCGA data is organized into three separate datasets: **TCGA_bioclin_v0** contains clinical and other metadata; **TCGA_hg19_data_v0** contains the original TCGA molecular data, which was originally generated based on the GRCh37/hg19 reference; and **TCGA_hg38_data_v0** contains the newer GRCh38/hg38-based data now available at the NCI-GDC.

All of the tables include one or more of the following identifiers which can be used for performing cross-table JOINs: `case_barcode`, `sample_barcode`, and `aliquot_barcode`. (Note that these were previously called `ParticipantBarcode`, `SampleBarcode`, and `AliquotBarcode`.) In addition, most tables also containa a `project_short_name` field (formerly called `Study`, *eg* TCGA-LUAD, TCGA-BRCA, *etc*).

(Note that in an attempt to be consistent with the NCI-GDC terminology, what we used to call a *project* is now called a *program* (*eg* TCGA, TARGET, CCLE, *etc*), while what was formerly known as a *study* is now called a *project* (and has also been prepended with the *program* name, so that `LUAD` has become `TCGA-LUAD`, *etc*).

Each dataset and table described below is linked directly the corresponding view in the BigQuery web UI where you can see the schema and other additional information for each table, preview its contents, *etc*.

- isb-cgc:TCGA_bioclin_v0:

- Clinical: This table contains one row for each TCGA case (aka patient or participant) with *any* available clinical information – over 11,000 cases are represented. Any given field in this table may be `null` for many patients, depending on tumor-type or data-availability. For example, the field `tobacco_smoking_history` is available for only about 3,000 patients.

- Biospecimen: This table is a *sample-centric* table, and contains one row of information for each of the (over 23,000) TCGA samples. Any given field in this table may be `null` for many samples, depending on the sample-type or the tumor-type.

- Annotations: This table contains annotations and related information obtained from the TCGA Annotations Manager (formerly available at the TCGA DCC).

- isb-cgc:TCGA_hg19_data_v0:

- Copy_Number_Segment_Masked: This table contains all available Copy Number segmentation data across all TCGA samples. Each row in the table describes a single copy-number segment for a single aliquot. The fields `chromosome`, `start_pos`, and `end_pos` specify the chromosomal coordinates (1-based) for the segment, the `num_probes` field specifies the number of probes on the SNP chip that went into estimating the mean copy-number for this segment, and finally the `segment_mean` provides the `log2(CN/2)` mean value estimate.

> Values near 0 represent "normal" copy-number, while larger positive values indicate *amplifications* and negative values indicate *deletions*.

- **DNA_Methylation**: This table contains **all** of the DNA methlyation data for all TCGA samples assayed on either the HumanMethylation 27k or 450k platforms. Please note that this is a very **large** table (with close to 4 billion rows), so query it with caution – a *single* query will cost *your* GCP project $2-3. Each row contains the methylation "beta" for a particular aliquot at a particular probe. Details about a particular probe, based on the `Probe_Id` field value (*eg* cg03879918) can be obtained from the `methylation_annotation` table (available in the isb-cgc:platform_reference dataset). For convenience, this data is also available in 24 chromosome-specific tables so that more targeted queries will need to scan less data (and will therefore be cheaper).

- **miRNAseq_Expression**: This table contains **all** of the miRNAseq stem-loop expression data for all TCGA samples assayed on either the Illumina GA or Illumina HiSeq platforms.

- **miRNAseq_Isoform_Expression**: This table contains **all** of the miRNA isoform-level expression (aka isomiR) data for all TCGA samples assayed on either the Illumina GA or Illumina HiSeq platforms.

- **Protein_Expression**: This table contains protein expression quantification estimates based on the RPPA (reverse phase protein array) platform. Note that only a subset (~70%) of the TCGA tumor samples were assayed on this platform. This technology uses antibodies which bind (sometimes non-specifically) to the target protein. In certain cases, an antibody may target a specific phosphorylated protein. Each row in this table includes an estimate of the `protein_expression`, with the following fields specifying the protein: `gene_name` (aka symbol), `protein_name`, `protein_base_name`, and `phospho`. Additional fields include the `antibody_source` and `validation_status`.

- **RNAseq_Gene_Expression_UNC_RSEM**: This table contains gene expression data from 10,289 samples assayed on the Illumina HiSeq platform and 818 samples assayed on the Illumina GA platform, all of which were then processed through the UNC "RNASeqV2" RSEM pipeline. Each row in this table contains the RSEM expression estimate for a single gene in a single aliquot. The gene symbol can be found in the fields `original_gene_symbol` (as originally given in the file submitted by UNC to the TCGA DCC), and `HGNC_gene_symbol` (the most current HGNC-approved gene symbol at the time this table was created). More details about specific genes can be obtained from any of the `GENCODE` tables available in the genome_reference dataset.

- **Somatic_Mutation_DCC**: This table contains all somatic mutations called across all TCGA tumor samples, based on aggregating all of the MAF files available at the DCC as of June 2016. Each mutation call was annotated using Oncotator, and many (though not all) of the resulting annotation fields were included in this table. Since multiple MAF files are sometimes available for a single tumor type, the MAF ETL process included steps to filter out duplicate mutation calls.

- **Somatic_Mutation_MC3**: This table is based on the unified "MC3" TCGA call set recently published by the TCGA Network. For more details or the original source file, please refer to Synapse. The original input file contained 114 columns but many were empty or duplicates of other columns. This table contains 75 columns. Additional details can be found in the table schema.

- **isb-cgc:TCGA_hg38_data_v0**: This dataset by and large mirrors the `TCGA_hg19_data_v0` dataset, and is based on the GRCh38/hg38 data now available from the NCI-GDC. In some cases the new data has been re-aligned to the new genome (in the case of any DNAseq or miRNA/mRNAseq based data), or the coordinates have been "lifted over" from hg19 to hg38 (in the case of probe/array-based data such as the SNP6/copy-number and the DNA Methylation data).

A set of reference data tables have also been created in BigQuery which you may find helpful when analyzing the TCGA data.

TARGET Clinical, Biospecimen and Molecular Data

The TARGET data is organized into two separate datasets: **TARGET_bioclin_v0** contains clinical and other metadata; and **TARGET_hg38_data_v0** contains the GRCh38/hg38-based data now available at the NCI-GDC.

All of the tables include one or more of the following identifiers which can be used for performing cross-table JOINs: `case_barcode`, `sample_barcode`, and `aliquot_barcode`. In addition, most tables also contain a `project_short_name` field (formerly called `Study`, *eg* TARGET-AML, *etc*).

Each dataset and table described below is linked directly the corresponding view in the BigQuery web UI where you can see the schema and other additional information for each table, preview its contents, *etc*.

- isb-cgc:TARGET_bioclin_v0:

- Clinical: This table is contains one row for each TARGET case (aka patient or participant) with *any* available clinical information – over 5,000 cases are represented. Note that most of these cases do not *yet* have molecular data available in BigQuery.

- Biospecimen: This table is a *sample-centric* table, and contains one row of information for each of the (over 7,000) TARGET samples.

- isb-cgc:TARGET_hg38_data_v0: This dataset will by and large mirror the `TARGET_hg38_data_v0` dataset, and is based on the GRCh38/hg38 data now available from the NCI-GDC. In some cases the new data has been realigned to the new genome (in the case of any DNAseq or miRNA/mRNAseq based data), or the coordinates have been "lifted over" from hg19 to hg38 (in the case of probe/array-based data such as the SNP6/copy-number and the DNA Methylation data).

- miRNAseq_Isoform_Expression: This table contains **all** of the miRNAseq isoform-level expression (aka isomiR) data *currently available* from the NCI-GDC.

- RNAseq_Gene_Expression: This table contains gene expression data from 481 samples (434 cases). Each row in this table contains the HTSeq expression estimates for a single gene in a single aliquot. The gene symbol can be found in the field `gene_name` and the Ensembl ID can be found in the `Ensembl_gene_id` and `Ensembl_gene_id_v` fields.

Additional Metadata

Additional related metadata is organized into the following datasets:

- isb-cgc:metadata: This dataset currently contains two tables which contain metadata about two additional TCGA data types: pathology and radiology images. More information about these image datasets can be found on the TCGA-images documentation page.

- isb-cgc:GDC_metadata: This dataset contains several tables which contain metadata describing the cases and files at the NCI-GDC, in both the legacy and the current data archives.

- isb-cgc:tcga_seq_metadata: This dataset contains several tables with metadata about the original hg19 sequence data (including both BAM and FASTQ files). The important common identifiers to link these tables back to other information is the `CGHubAnalysisID` (which sometimes may be written `CGHub_analysisID`). In alphabetical order by name, these tables are:

- isb-cgc:tcga_cohorts: This dataset contains a series of curated cohorts, one for each of the 33 TCGA tumor types, named according to the tumor abbreviation, *eg* BRCA. A "cohort" is defined as a paired list of case- and sample-barcodes. In order to be included, molecular data from at least one of the main platforms must be available for that sample, and there must be no disqualifying annotation for that sample or the case (aka patient). For example, the BRCA cohort table contains 1086 unique cases and 2221 unique samples, but a query of the Clinical table for all BRCA cases will return 1097 cases, and a similar query of the Biospecimen table for all BRCA samples will return 2302 samples. The Annotation table contains annotations of one type or another for

122 "entities" in the TCGA-BRCA proejct affecting 33 BRCA cases, 2 BRCA samples, 18 BRCA analytes, and 69 BRCA aliquots.

ETL (Extract, Transform, Load) Details

The data in the BigQuery tables is generally identical to the information that can also be obtained from the NCI-GDC, but for users interested in the nitty-gritty details, information is provided here about the ETL (extract, transform and load) steps that were performed for each of the data types.

Before we go into data-type-specific details, a few general notes on formatting and data curation:

- All data uploaded into ISB-CGC BigQuery tables use a consistent UTF-8 character set. If the encoding of a character from the original file could not be detected, that character was ignored. Character encodings were detected using the Python library Chardet.

- All missing information value strings such as: `none`, `None`, `NONE`, `null`, `Null`, `NULL`, , `NA`, `\_\_UNKNOWN\_\_`, `<blank>` , and `?`; are represented as NULL values in the BigQuery tables (or may not appear at all, depending on the table schema).

- Numbers are stored as integer or floating point values. The original ASCII files sometimes used scientific notation or included comma separators, but these are not preserved in the BigQuery tables.

- End of File (EOF) and End of Line (EOL) delimiters, including CTRL-M characters, were all removed when the raw files were originally parsed.

- Single and double quotes around the values were removed, but in cases where there were quotation marks within a string, they were not removed.

- Whenever necessary, the SDRF file (in the mage-tab archive associated with each data archive) was parsed to find the correct association between the aliquot barcode and the Level-3 data file(s).

Data-Type Specific ETL Details

Clinical

The Clinical table contains one row per TCGA participant (aka patient or donor). Each TCGA participant is uniquely represented by a TCGA barcode of length 12, *eg* `TCGA-2G-AAM4`. (For more information on how TCGA barcodes were created and how to *"read"* a TCGA barcode, click on the preceding link.)

Clinical Feature Selection

In the first pass, any XML features with the tag `procurement\_status=Completed` which were found to exist in at least 20% of the participants in any one Disease (aka tumor-type) were considered for selection. A few important features related to smoking, pregnancy, *etc* were added to the list during a manual-curation pass.

Selected fields from the both the clinical, auxiliary, ssf, and omf XML files were then extracted and loaded into the BigQuery table.

Additionally, only the most recent follow-up information was included (for cases where multiple follow-up sections existed in the clinical XML file).

XML Parsing

Each clinical XML file is divided into `admin` and `case` blocks, and each of these were processed separately.

While iterating through the case block of information, all elements (XML tags) and their values were collected. For `follow-up` blocks, only the most recent (based on sequence number) sub-block elements were kept.

In the final pass, case elements and follow-up elements were carefully merged with preference given to follow-up elements.

Transforms

Different survival-related fields are completed based on the value of the `vital_status` field:

- for all patients with `vital_status=Alive`:
    - days_to_last_known_alive should not be NULL
    - days_to_last_known_alive is set to days_to_last_followup
    - days_to_death is set to NULL
- for all patients with `vital_status=Dead`:
    - days_to_death should not be NULL (if it is NULL, and days_to_last_followup is not NULL, then vital_status is set to "Alive"
    - days_to_last_known_alive is set to days_to_death
    - days_to_last_followup is set to NULL
- `pregnancies` and `total_number_of_pregnancies` were merged into a single `pregnancies` field. Counts above four are represented as 4+ (e.g: [0,1,2,3,4+])
- `number\_of\_lymphnodes\_examined` and `lymph\_node\_examined\_count` were merged into a single `number\_of\_lymphnodes\_examined` field
- **country and country_of_procurement were merged into a** single `country` field

The following fields were extracted from the ssf XML file:

- `histological\_type`
- `country`
- `other\_dx`
- `tobacco\_smoking\_history`
- `gleason\_score\_combined`
- `history\_of\_neoadjuvant\_treatment`

The following fields were extracted from the omf XML file:

- `other\_malignancy\_malignancy\_type`
- `other\_malignancy\_anatomic\_site`
- `other\_malignancy\_histological\_type`

When an auxiliary XML file exists for a participant, and the batch numbers in both the clinical XML and the auxiliary XML file match, the following fields are extracted from the auxiliary XML file and added to the Clinical table:

- `hpv\_calls,`
- `hpv\_status,`
- `mononucleotide\_and\_dinucleotide\_marker\_panel\_analysis\_status,`

Finally, the patient BMI was calculated based on the `height` and `weight` values (when both were present) and was added to the Clinical table.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Biospecimen

The Biospecimen table contains one row per TCGA sample. Each TCGA sample is uniquely represented by a TCGA barcode of length 16, *eg* `TCGA-2G-AAM4-10A`. (For more information on how TCGA barcodes were created and how to *"read"* a TCGA barcode, click on the preceding link.)

XML Parsing

The TCGA data at the DCC exists in XML files which have been uploaded into Google Cloud Storage. Selected fields from these XML files were then extracted and loaded into the "Biospecimen" table in BigQuery.

Some of the biospecimen values in the XML files are available on a per-slide and/or per-portion basis, and these have been aggregated and averaged. The number of slides and the number of portions per sample is also included in the table.

Filters

- Samples for which `is\_ffpe=True` were removed.
- Patients or Samples for which `Project` value was *not* TCGA were removed.

The following fields were extracted from the ssf XML file:

- `days\_to\_sample\_procurement`
- `tissue\_anatomic\_site`
- `tissue\_anatomic\_site\_description`
- `tissue\_anatomic\_site`

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Somatic DNA Mutations

The Somatic Mutations tables, Somatic_Mutation_DCC, Somatic_Mutation_MC3, and Somatic_Mutation in BigQuery contain somatic mutation calls collected from the open-access MAF

files from 30 tumor types.

For each MAF file, some simple data-cleaning performed, it was then annotated using Oncotator and then further processed to remove duplicates before being merged into a single table.

Data-Cleaning

- Remove any lines where the `build` is not `37`

- Remove any lines where the `chr` is not in `[1-22, X, Y]`

- Remove any lines where the `Mutation_Status` is not `Somatic`

- Remove any lines where the `Sequencer` is not an Illumina platform

- Change the column labels to match what Oncotator expects (*eg* `ncbi_build` becomes `build`, `chromosome`, `chr`, *etc*.

Oncotator Annotation

Each file was then annotated using Oncotator version 1.5.1, with the `Jan2015` database, and the options `--input_format=MAFLITE --output_format=TCGAMAF`.

The outputs of Oncotator were lightly processed to change the column labels and to remove certain special characters from strings.

Duplicate Removal

Because many tumor types have several "current" MAF files and deciding which one is the "best" is a non-trivial process, and also because some tumor samples may have had mutations called relative to a tissue normal and also relative to a blood normal, it is possible that the same mutation has been called multiple times. In order to eliminate over-counting of mutations, we sought to remove these duplicate calls from the result of concatenating all of the annotated MAF files using the following rules:

- if a mutation in the same position is called in a particular tumor sample with respect to multiple matched normals, we prefer the "blood derived normal" over the "solid tissue normal"

- if a mutation in the same position is called in multiple aliquots for one tumor sample, we prefer the "D" analyte over the "W" analyte (*eg* `TCGA-B0-5695-01A-11D-1534-10` over `TCGA-B0-5695-01A-11W-1584-10`)

- if both aliquots are "D" (or both are "W") analytes, then we choose based on the data-generating-center (the final two characters in the aliquot barcode), preferring first:

  - `01`, `08`, or `14` (all of which refer to `broad.mit.edu`)

  - `09`, `21`, or `30` (all of which refer to `genome.wustl.edu`)

  - `10` or `12` (both of which refer to `hgsc.bcm.edu`)

  - `13` or `31` (both of which refer to `bcgsc.ca`)

  - `18` or `25` (both of which refer to `ucsc.edu`)

- finally, in the event that a mutation in the same position was called by the same center, with the same type of matched normal, and the same type of analyte, then we choose the aliquot with the larger value in the final 4-digit sequence in the barcode (positions 21:25)

In addition, any exact duplicates (*ie* all fields describing a mutation are the same) in the merged file are removed, and the final result uploaded into BigQuery.

The result are tables containing over 5.8 million mutations called on 8435 tumor samples from 8373 patients.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

DNA Copy-Number Segments

The Copy_Number_segments table contains one row *per* copy-number segment *per* TCGA aliquot. Each TCGA aliquot is uniquely represented by a TCGA barcode of length 24, *eg* `TCGA-04-1517-01A-01D-0533-01`. (For more information on how TCGA barcodes were created and how to *"read"* a TCGA barcode, click on the preceding link.)

There is also a GDC Copy_Number_segments table that has been reprocessed against the HG38 genomic build.

Platform

DNA Copy-Number data was generated for the TCGA project using the Affymetrix GenomeWide Human SNP 6.0 Array.

Pipeline

DNA Copy-Number data was generated for the TCGA project at the Broad Genome Characterization Center. A `DESCRIPTION.txt` file is included with each data archive at the DCC describing the algorithms, methods, and protocols used to produce the Level-1, Level-2, and Level-3 data.

ETL Details

Each Level-3 data archive contains 4 output files per sample assayed: two based on the hg18 reference, and two based on the hg19 reference for the TCGA HG19 data table. The BigQuery table is populated only with the files ending with `nocnv\_hg19.seg.txt`. The `num_probes` and `segment_mean` fields in the raw files are sometimes represented using Exponential Scientific Notation (*eg* 8.7E+07) and were interpreted as integer or floating-point values respectively.

The mapping between TCGA aliquot barcodes and Level-3 data files was obtained from the SDRF file.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

DNA Methylation

The DNA Methylation data for genomic build HG19 DNA Methylation HG19 table contains one row per CpG probe and TCGA aliquot. The DNA Methylation data for genomic build HG38 is pulled from the GDC, DNA Methylation HG38.

Each TCGA aliquot is uniquely represented by a TCGA barcode of length 24, *eg* `TCGA-04-1517-01A-01D-0533-01`. (For more information on how TCGA barcodes were created and how to *"read"* a TCGA barcode, click on the preceding link.)

The platform annotation information needed to analyze this data is also available in a BigQuery table. For more information, see the Reference Data section of this documentation.

Platform

DNA Methylation data was generated for the TCGA project using the Illlumina HumanMethylation27 BeadChip and its successor, the HumanMethylation450 BeadChip.

Pipeline

DNA Methylation data was generated for the TCGA project at the JHU-USC genome characterization center. A `DESCRIPTION.txt` file is included with each data archive at the DCC describing the algorithms, methods, and protocols used to produce the Level-1, Level-2, and Level-3 data.

ETL Details

The BigQuery table is populated only with the files matching the pattern `%.HumanMethylation%.txt`. The data from both 27k and 450k platform have been merged together into a single table. A few samples were run on both platforms, and for those samples, the 450k data takes precedence. The table includes a platform column indicating the source of each data value.

In addition:

- any CpG probes for which the Level-3 `Beta_Value` is NA or NULL, are left out
- only the `Probe_Id` and `Beta_Value` fields from the Level-3 data files are stored in the BigQuery table

Since the DNA_Methylation_betas table is so large, we also provide chromosome-specific tables that can be used for faster queries.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

mRNA Expression

Although gene expression data for the TCGA project was produced by two different centers, only the data produced by the UNC RSEM pipeline was incorporated into the GDC. The GDC has also reprocessed that data against HG38 resulting in a second BigQuery table.:

- the data produced by the UNC LCCC and the resulting normalized RSEM values are stored in one table
- and the data produced by the GDC and the resulting normalized RPKM values are stored in another table

UNC RNAseqV2 Pipeline

A `DESCRIPTION.txt` file describing the algorithms, methods, and protocols used to produce the Level-1, Level-2, and Level-3 data can be obtained from the TCGA DCC.

The BigQuery table was populated using the values in files matching the pattern

`%.rsem.genes.normalized\_results`. These raw "RSEM genes normalized results"

files have two columns, both of which are stored in the BigQuery table. The first column contains the `gene_id` which contains two parts separated by a `|`, *eg*: `TP53|7157`. The second column contains the `normalized_count` representing the expression value for that gene.

The `gene\_id` column is split into two components and stored as separate columns: `original\_gene\_symbol` and `gene\_id`. Based on the `gene_id`, the current HGNC approved gene symbol is looked up and added as a third column: `HGNC_gene_symbol`.

GDC RNAseq Pipeline

The original DCC genomic data was reprocessed against the `HG38` genomic build.

The `gene` string is split into two or three components and stored as separate columns: `original_gene_symbol` and `gene_id` and, if there is a third component, a `gene_addenda` column. If one component is simply `?`, that character string is replaced by a `NULL` value. Finally, the current HGNC approved gene symbol is looked up and added as an additional column: `HGNC_gene_symbol`.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

microRNA Expression

The current ISB TCGA data pipeline uses a Perl script `expression\_matrix\_mimat.pl` provided by BCGSC which reads the isoform data files and outputs expression values for "mature microRNAs". This output matrix contains a consistent number of mature microRNAs, referred to using a combination of the microRNA gene name and the unique accession number, eg: "hsa-mir-21.MIMAT0000076". During ETL, this string is split into two parts and stored as separate columns in the BigQuery table. The entire matrix is then melted into a flat structure (known as the tidy data format) and loaded into the table.

Only the isoform files matching the pattern `%.hg19.mirbase20.isoform.quantification.txt` and containing hg19 data were used. The aliquot barcode information was obtained from the SDRF file associated with the Level-3 isoform data file.

There is also a table with the expression data and corresponding tables reprocessed by the GDC against the HG38 genomic build for isoform and expression

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Protein Expression (RPPA)

The raw protein data file contains just two columns: The "Composite Element REF", which corresponds to the third column in the antibody annotation file, and the estimated expression value for that particular protein. The "Composite Element REF" was parsed to generate additional

information(see details in the formatting section). The BigQuery tables HG19 Protein_Expression and HG38 Protein_Expression

was populated with all TCGA Level-3 RPPA data matching the pattern - "%_RPPA_Core.protein_expression%.txt".

The antibody annotation files are parsed to get the relationship between the antibody name and the associated proteins, and genes. Below is the detailed explanation about the generation of the antibody, gene, protein map.

Generation of Composite_element_ref, gene, and protein name map

(Manual Curation of the gene and protein names)

- Check the antibody annotation files for missing columns.
- If 'protein_name' is missing, generate one from 'composite_element_ref'
- Make a map of 'composite_element_ref',' gene_name', 'protein_name' values.

- Check any other variant of the gene and protein symbols in the table.

- HGNC Validation

- If the gene symbol is in the HGNC approved symbols, 'Approved'. Gene_symbol = Gene_symbol.

- If not, check the Alias symbols. If found, Gene_symbol = Alias_symbol.

- If not, check the Previous symbols. If found, Gene_symbol = 'Approved' Gene_symbol.

- If not, Gene_symbol = Gene_symbol

- The file generated is manually curated and fed back into the algorithm.

Formatting

- Duplicate the rows if there are multiple genes concatenated in the "gene_name" value. For example: 'gene_name' with value like 'AKT1 AKT2 AKT3' is stored as three separate rows with each gene in a row.

- 'Protein_Name' is split into 'Protein_Basename', Phospho' and are stored as separate columns.

- 'Composite element ref' is parsed to get 'validationStatus' and 'antibodySource' - both are stored as separate columns in the BigQuery table.

- Data from both Illumina GA and HiSeq platforms are stored in the same table.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Annotations

The TCGA Annotations BigQuery table was created based on the contents of the JSON file obtained from the TCGA Annotation manager Web Service API. The deeply nested JSON file was first flattened, and then a subset of the fields were selected to be loaded into the BigQuery table. In the flattening process, sub-level field names were prefixed with the parent name, separated by an underscore. These names have now been updated to reflect their names in the Genome Data Commons (GDC) Annotations API. Please refer directly to BigQuery for the table schema.

Original field name	New field name
items_disease_abbreviation	project_short_name
items_item	entity_barcode
items_itemType_itemTypeName	entity_type
annotationCategory_categoryName	category
annotationCategory_annotationClassification_annotationClassificationName	classification
notes_noteText	notes
notes_dateAdded	date_created
notes_dateEdited	date_edited
not available	case_gdc_id
not available	case_barcode
not available	sample_barcode
not available	aliqout_barcode

Sample and Participant barcodes are filled in (*ie* not null) whenever the "entity_barcode" is at least 16 or 12 characters long, respectively. For example, a "Shipped Portion" would result in a filled in "case_barcode" and "sample_barcode" fields. Please note, however, that the annotation applies *only* to the item specified in the "entity_barcode" field, the *type* of the item is specified in the "entity_type" field with the following caveat. If an annotation is on the case, then it

---

applies to all its samples, if on a sample, to all its portions but does not apply to other samples for that case, and so on down to the aliquot, which only applies to that aliquot.

Please note that the TCGA annotations available at the NCI Genomic Data Commons may differ from those found in this BigQuery table. Please let us know if you have questions or concerns about the contents of this table.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Data in Cloud Storage

At this time, all controlled-access data files are stored in Google Cloud Storage (GCS) in their original form, as obtained from the data repository. This includes these major data types and formats:

- RNA-Seq **FASTQ** files (unaligned reads, typically compressed tar-files)
- DNA-Seq and RNA-Seq **BAM** files (aligned reads)
- Genome-Wide SNP6 array **CEL** files
- Variant-calls in **VCF** files

In order to access these controlled data, a user of the ISB-CGC must first be authenticated by NIH (via the ISB-CGC web-app). Upon successful authentication, the users's dbGaP authorization will be verified. These two steps are required before the user's Google identity is added to the access control list (ACL) for the controlled data. At this time, this access must be renewed every 24 hours.

Summary of Data Available in GCS

Format	Data Type	# of Files	Total Size
BAM	DNA-Seq	73487	1407 TB
BAM	RNA-Seq	47818	216 TB
FASTQ	RNA-Seq	13207	91 TB
CEL	DNA (SNP6)	22529	1.6 TB
VCF	DNA-Seq	47319	0.5 TB

Working with data in GCS

Working with large-scale data hosted by the ISB-CGC in Google Cloud Storage requires some familiarity with tools such as the Google Cloud SDK, Google Compute Engine, Virtual Machines and Docker.

Please see our DIY Workshop and in particular the section on "Computing in the Cloud" for additional references and tutorial material.

Our metadata tables in BigQuery can be used to explore the available data and choose which BAM files you're most interested in working with – before you take on processing an entire petabyte of data! Feel free to email us at info@isb-cgc.org with questions.

BAM-slicing in the Cloud

BAM files can vary in size from close to 1 TB down to 1 MB, and frequently a researcher is only interested in extracting a small slice of the entire sequence. This is referred to as "BAM-slicing" and the latest release (1.4) of the htslib library adds the capability to perform BAM-slicing directly on BAM files in GCS to widely used tools such as samtools. (You will need to build with `--enable-libcurl` to enable support for access to data both in GCS and S3.) This new functionality allows you to run, for example:

```
$ ./samtools view gs://gdc-ccle-open/0a109993-2d5b-4251-bcab-9da4a611f2b1/C836.Calu-3.
→2.bam 7:140453130-140453140

Index file location: gs://gdc-ccle-open/96b56036-b278-45cd-a7ca-265f589ff951/C836.
→Calu-3.2.bam.bai
```

If you want to access a controlled-access BAM file, you'll need to provide credentials first:

```
$ export GCS_OAUTH_TOKEN=`gcloud auth application-default print-access-token`
```

If you run into problems, it's a good idea to verify that you have the correct url and also that you have access to this file by using the gsutil command-line tool from the cloud SDK:

```
$ gsutil ls -l gs://gdc-ccle-open/0a109993-2d5b-4251-bcab-9da4a611f2b1/C836.Calu-3.2.
→bam

$ gsutil ls -l gs://gdc-ccle-open/96b56036-b278-45cd-a7ca-265f589ff951/C836.Calu-3.2.
→bam.bai
```

Other Options for BAM-slicing

The NCI-GDC has also implemented a BAM-slicing API on top of their data repository. This API can be accessed programmatically as documented here or interactively on any of the file-specific data-portal pages like this one for a TCGA-BRCA whole-exome BAM file. (The "BAM Slicing" button is in the upper right corner of the page.)

The GA4GH API provides another option to BAM-slicing, and has been implemented by Google on top of the database-backed Google Genomics technology. You can find more information about the GA4GH API here with information about some open-access data hosted by the ISB-CGC which you are welcome to experiment with.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Data in Google Genomics

Google Genomics was a database-backed technology that allowed users to query reads and variants using the GA4GH API.

On December 7th, 2017 Google announced that the "reads API" has been replaced by the htsget protocol , and the "variants API" has been replaced by htsget and the new open-source "Variant Transforms" a tool for uploading variant data directly into BigQuery.

In 2017, the GA4GH team introduced the **htsget protocol** to allow users to download read data for subsections of the genome in which they are interested. This is a richer and more flexible approach to working with reads data. It allows you to keep your genomics data in a common BAM file format on Google Cloud Storage and work with it efficiently from your computation pipelines, using standard bioinformatics tools. Google has already launched their own open source implementation of this protocol, which you can use to access your reads data. Many popular tools such as samtools and htslib have been updated by the community to support htsget. Documentation is provided here.

After analyzing usage of the Variants API, Google found that users primarily used it to import variant data and then export it to BigQuery. To save time and effort, Google created **Variant Transforms**, an open source tool for directly importing VCF data into BigQuery. Variant Transforms and its documentation is published here. Variant Transforms is more scalable than the legacy Variants API, and it has a robust roadmap with a dedicated Google team.

Because of the new APIs described above, ISB-CGC has removed the CCLE DNA-Seq and RNA-Seq data previously hosted in Google Genomics. Please don't hesitate to reach out to us at feedback@isb-cgc.org if there are issues you would like to discuss or get help with. Google also has a set of resources to follow on GitHub.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Reference Data

ISB-CGC Hosted Reference Data

In order to facilitate working with the TCGA data tables that the ISB-CGC is hosting in BigQuery, additional reference data tables have also been created, others are hosted by Google Genomics, and suggestions for more are welcome at feedback@isb-cgc.org.

Genome Reference Data

Reference data that describes or annotates the human (or other) genome(s) is described in this section. Reference data hosted by the ISB-CGC in BigQuery tables are available in the `isb-cgc:genome_reference` dataset. Tables based on gene-sets such as Ensembl and GENCODE can be used to find the genomic coordinates and identifiers for genes of interest, in order to perform queries that join tables with gene-symbol based data to tables with genomic-coordinate based data or tables that use other gene identifiers, for example.

For additional details about each of these tables, please use the BigQuery web UI to access each of these tables and look at the information on the **Details** page. (Look for the Details button between the Schema and Preview buttons, beneath the table name.)

- **Ensembl**
    - GRCh37 : Release 75, the final build of the Ensembl gene-set mapped to GRCh37
    - GRCh38 : Release 87, the most recent Ensembl gene-set mapped to GRCh38
- **GENCODE**
    - GRCh37 : Release 19, the final build of the GENCODE gene-set mapped to GRCH37
    - GRCh38 : Releases 22, 23, and 24 from GENCODE are all available (because the TCGA data has been reprocessed by at least one center using each of these three different releases)
- **Gene Ontology Consortium** : Tables based on GO annotations and the GO ontology.

- **Kaviar** : The latest hg19- and hg38-based Kaviar databases are available. Kaviar is a compilation of SNVs, indels, and complex variants observed in humans, designed to facilitate testing for the novelty and frequency of observed variants.

- **liftOver_hg19_to_hg38** : This table provides a mapping of each hg19 position to the corresponding position in hg38, and can be used to perform a liftOver operation in BigQuery

- **miRBase**

    - GRCh37 : The human portion of version 20 of the miRBase database; including genomic coordinates for human microRNAs.

    - GRCh38 : The human portion of version 21 of the miRBase database; including genomic coordinates for human microRNAs.

- **miRTarBase** The recently updated miRTarBase database (release 6.1)

- **Reactome**

    - Ensembl2Reactome

    - miRBase2Reactome

Platform Reference Data

Some reference data is necessary in order to work with data generated by specific platforms such as the Illumina DNA Methylation array, or the Affymetrix Genome-Wide Human SNP Array 6.0. This section will provide links to existing sources of information elsewhere on the web, or will describe additional resources that are hosted by the ISB-CGC. If there are additional platform reference sources that you would like to see hosted in BigQuery tables, please let us know at feedback@isb-cgc.org.

- **DNA Methylation Platform**

    - Most of the DNA Methylation data produced by the TCGA project was obtained using the Illumina Infinium HumanMethylation450 (aka 450k) BeadChip array. Some of the earlier tumor types were assayed on the older, 27k array.

    - Although additional details can be found at the Illumina webpage, we have uploaded the platform annotation information into the BigQuery table `isb-cgc:platform_reference.methylation_annotation`

    - Each CpG locus is uniquely identified as described in this technical note and this unique identifier can be used to look up and cross-reference data between the TCGA DNA methylation data table and the platform annotation table.

    - The original Illumina-provided CpG coordinates have been *"lifted over"* from hg19 to hg38

- **Genome-Wide SNP Array** - The technical documentation for the Affymetrix Genome-Wide Human SNP Array 6.0 array can be found here

Other Reference Data Sources

In collaboration with the Wellcome Trust Sanger Institute, the ISB-CGC is hosting the COSMIC database.

Google Genomics maintains a list of publicly available datasets, including **Reference Genomes**, the **Illumina Platinum Genomes**, information about the **Tute Genomics Annotation** table, *etc*.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Data Releases and Future Plans

Release Notes

- June 25, 2018 : GDC data release 12 was posted on Wednesday, June 13, 2018

    - There is absolutely no change in the legacy archive data between DR11 and DR12

    - There is also no change in the total number of cases in either archive

    - The number of files in the current archive has incrased from 329,165 to 356,381:

  - 67,220 files were removed

  - 94,436 files were added

More details about the changes to the current archive of TCGA data: - Copy Number Variation | Genotyping Array | TXT files:

  - 22376 Copy Number Segment files replaced (ie removed and added)

  - 22376 Masked Copy Number Segment files replaced

  - Biospecimen | BCR XML files:

  - 11294 files replaced

  - Clinical | BCR XML files:

  - 11160 files removed / 11167 files added (ie 7 extra files)

  - Biospecimen | Diagnostic Slide | SVS files:

  - 11730 Slide Image files added

  - Biospecimen | BCR SSF XML files:

  - 10557 Biospecimen Supplement files added

  - Biospecimen | BCR Auxiliary XML files:

  - 2884 Biospecimen Supplement files added

  - Clinical | BCR OMF XML files:

  - 1051 Clinical Supplement files added

  - Biospecimen | BCR Biotab files:

  - 340 Biospecimen Supplement files added

  - Clinical | BCR Biotab files:

  - 226 Clinical Supplement files added

  - Simple Nucleotide Variation | WXS | VCF | Varscan2 files :

  - 1 Raw Simple Somatic Mutation file removed (2017-03-04)

  - 1 Annotated Somatic Mutation file removed (2017-06-17)

  - both for ESCA samples: TCGA-VR-A8ET-01A-11D-A403-09;TCGA-VR-A8ET-10B-01D-A403-09

For TARGET data: - RNA-Seq data:

  - 3 BAM files and 9 Gene Expression Quantification files removed

- sample barcodes: TARGET-30-PAKYZS-01A-01R, TARGET-30-PAMEZH-01A-01R, TARGET-30-PANRRW-01A-01R

- Raw CGI Variant | WGS | Combined Nucleotide Variation | VCF files:

- 435 files added

- June 4, 2018: metadata tables for GDC data release 11 are now available in BigQuery

- May 8, 2018: the gnomAD database (release 2.0.2, dated October 2017) is now available in BigQuery! **isb-cgc:genome_reference.gnomAD_20171003_GRCh37**

- April 30, 2018: recently released (2018-04-01) ClinVar VCFs are now available in BigQuery! two new tables (**ClinVar_20180401_GRCh37** and **ClinVar_20180401_GRCh38**) can be found in our genome_reference dataset; also available is dbSNP build 151 (announced 2018-04-24): **isb-cgc:genome_reference.dbSNP_b151_GRCh37p13_All**

- February 22, 2018: a genenames_mapping table has been added to our numerous reference sources in BigQuery to simplify mapping between HGNC IDs, HGNC symbols, Entrez Gene IDs, Ensembl Gene IDs, Pubmed IDs, and RefSeq IDs!

- June 9, 2018: metadata tables for GDC data release 10 are now available in BigQuery

- May 8, 2018: release 85 of the COSMIC database is now available in BigQuery

- February 13, 2018: release 84 of the COSMIC database is now available in BigQuery

- December 19, 2017: The ISB-CGC cohort metadata has been update to reflect the new and update TARGET gene expression data provided by the GDC in their data release 9.

- December 6, 2017: the GDC release 9 included some updated and new TARGET gene expression data. The BigQuery table **isb-cgc:TARGET_hg38_data_v0.RNAseq_Gene_Expression** has been updated to reflect this.

- November 7, 2017: release 83 of the COSMIC database is now available in BigQuery

- November 3, 2017: metadata tables for GDC data release 9 are now available in BigQuery

- October 30, 2017: the 'harmonized' hg38 TCGA VCF files (raw and annotated) are now available in the ISB-CGC controlled-data repository in Google Cloud Storage

- August 30, 2017: hg38 TARGET VCF files (raw and annotated) are now available in the ISB-CGC controlled-data repository in Google Cloud Storage

- August 3, 2017: release 82 of the COSMIC database is now available in BigQuery

- June 30, 2017: hg19 and hg38 TARGET WXS, RNA-Seq, and miRNA-Seq BAM files are now available in the ISB-CGC controlled-data repository in Google Cloud Storage

- May 9, 2017: release 81 of the COSMIC database is now available in BigQuery

- May 5, 2017: a table mapping between UniProtKB accessions and identifiers has been added to our reference dataset: **isb-cgc:genome_reference.UniProtKB_idmapping**

- April 10, 2017: We have re-organized our TCGA clinical, biospecimen, and molecular data into new datasets in BigQuery – please find them at isb-cgc:TCGA_bioclin_v0, isb-cgc:TCGA_hg19_data_v0, and isb-cgc:TCGA_hg38_data_v0. The hg19 data can also be found in the GDC's legacy archive, while the hg38 data is available at the GDC data portal.

- March 30, 2017: the 'harmonized' hg38 TCGA miRNA-Seq BAM files from the initial GDC data release are now available in the ISB-CGC controlled-data repository in Google Cloud Storage

- February 20, 2017: in collaboration with the Sanger Institute, the COSMIC database is now available in Big-Query (registered users only)

- February 5, 2017: genomic coordinates (in GFF3 format) for human microRNAs added for miRBase v20 and v21 to the **isb-cgc:genome_reference** BigQuery dataset

- January 30, 2017: the final, unified "MC3" TCGA somatic mutations call set is available in the BigQuery **isb-cgc:hg19_data_previews** dataset (also available on Synapse)

- January 10, 2017: **miRBase_v20** table added to the **isb-cgc:genome_reference** BigQuery dataset

- January 4, 2017: Ensembl gene-set releases 75 (GRCh37) and 87 (GRCh38) are now also available in the **isb-cgc:genome_reference** BigQuery dataset.

- December 30, 2016: the 'harmonized' hg38 TCGA WXS BAM files and RNA-Seq BAM files from the initial GDC data release (1.0), as well as the legacy hg19 TCGA 'Level 2' Genome-Wide SNP6 array genotype files ('birdseed') files are now available in the ISB-CGC controlled-data repository in Google Cloud Storage

- November 16, 2016: TCGA proteomics data from the CPTAC (Phase II) is now available in Google Cloud Storage

- November 14, 2016: TCGA radiology and tissue slide images are now available in Google Cloud Storage! This includes radiology images (DICOM files) from the Cancer Imaging Archive (TCIA) and tissue slide images from the NCI-GDC data portal (SVS files).

- September 10, 2016: **GENCODE** versions 19, 22, 23, and 24 are all now available in the **isb-cgc:genome_reference** BigQuery dataset, with an updated and more complete schema – note also that the naming convention is now **GENCODE_v19** rather than GENCODE_r19; also that v19 is the *last* version based on hg19/GRCh37, and all subsequent versions are based on hg38/GRCh38

- August 31, 2016: a table based on the latest liftOver hg19-to-hg38 chain files is available in the **isb-cgc:tcga_genome_reference** BigQuery dataset

- August 26, 2016: a set of tables based on running Picard over ~67,000 TCGA bam files in GCS have been added to the **isb-cgc:tcga_seq_metadata** BigQuery dataset: information contained in these tables includes bam-index stats, insert-size metrics, quality-distribution metrics, and quality-yield metrics – these tables can be used in conjunction with the FastQC-based tables to look for bam and/or fastq data files that meet your analysis criteria

- August 21, 2016: new **miRBase_v21** table added to the **isb-cgc:genome_reference** BigQuery dataset

- August 20, 2016: updated **hg19** and **hg38** Kaviar tables added to the **isb-cgc:genome_reference** BigQuery dataset

- August 17, 2016: new **isb-cgc:GDC_metadata** BigQuery dataset containing metadata for both *legacy* and *current* files hosted at the NCI-GDC.

- July 28, 2016: new **isb-cgc:tcga_201607_beta** BigQuery dataset based on the *final* TCGA data upload from the DCC. This dataset largely mirrors the previous **isb-cgc:tcga_20510_alpha** dataset and is now also supporting the ISB-CGC Web-App. The curated TCGA cohort tables in the **isb-cgc:tcga_cohorts** BigQuery dataset have also been updated.

- June 24, 2016: an updated listing of all ISB-CGC hosted data in Google Cloud Storage (GCS) is now available in the **GCS_listing_24jun2016** table in the **isb-cgc:tcga_seq_metadata** dataset in BigQuery, in addition the **CGHub_Manifest_24jun2016** table contains the final CGHub Manifest prior to the transition of all data to the Genomic Data Commons.

- June 18, 2016: new **GENCODE_r24** table added to the **isb-cgc:genome_reference** BigQuery dataset

- May 13, 2016: new **NCBI_Viral_Annotations_Taxid10239** table added to the **isb-cgc:genome_reference** BigQuery dataset

- May 9, 2016: new **Ensembl2Reactome** and **miRBase2Reactome** tables added to the **isb-cgc:genome_reference** BigQuery dataset

- **May 3, 2016: new isb-cgc:tcga_seq_metadata BigQuery dataset contains metadata and FastQC metrics for thousands of T**

- – **CGHub_Manifest** table contains metadata for all TCGA files at CGHub as of April 27th, 2016

  – **GCS_listing_27apr2016** table contains metadata for all TCGA files hosted by ISB-CGC in GCS

  – **RNAseq_FastQC** table contains metrics derived from FastQC runs on the RNAseq data files, including urls to the FastQC html reports that you can cut and paste directly into your browser

  – **WXS_FastQC** table contains metrics derived from FastQC runs on the exome DNAseq data files

- April 28, 2016: **GO_Ontology** and **GO_Annotations** tables added to the **isb-cgc:genome_reference** BigQuery dataset

- March 14, 2016: with the release of our **Web-App**, controlled-data is now accessible (programmatically) to users who have previously obtained dbGaP approval for TCGA data and go through the NIH authentication process built-in to the Web-App.

- February 26, 2016: new CCLE dataset in BigQuery **isb-cgc:ccle_201602_alpha** includes sample metadata, mutation calls, copy-number segments, and expression data (metadata includes full cloud-storage-path for world-readable BAM and SNP CEL files, and Genomics dataset- and readgroupset-ids for sequence data imported into Google Genomics)

- February 22, 2016: Kaviar database now available in the **isb-cgc:genome_reference** BigQuery dataset

- February 19, 2016: CCLE RNAseq and DNAseq bam files imported into **Google Genomics**

- January 10, 2016: **GENCODE_r19** and **miRBase_v20** tables added to the **isb-cgc:genome_reference** BigQuery dataset

- December 26, 2015: public release of new **isb-cgc:genome_reference** BigQuery dataset: the first table is based on the just-published **miRTarBase** release 6.1

- December 12, 2015: curated TCGA cohort lists available in **isb-cgc:tcga_cohorts** BigQuery dataset

- November 16, 2015: initial upload of data from CGHub into **Google Cloud Storage** (GCS) complete (not publicly released)

- **November 2, 2015: first public release of TCGA open-access data in BigQuery tables**

  – **isb-cgc:tcga_201510_alpha** dataset contains updated set of BigQuery tables, based on data available at the TCGA DCC as of October 2015

  – includes **Annotations** table with information about redacted samples, etc

  – **isb-cgc:platform_reference** contains annotation information for the Illumina DNA Methylation platform.

- October 4, 2015: complete data upload from TCGA DCC, including controlled-access data

- **September 21, 2015: draft set of BigQuery tables (not publicly released)**

  – **isb-cgc:tcga_201507_alpha** dataset containing clinical, biospecimen, somatic mutation calls and Level-3 TCGA data available at the TCGA DCC as of July 2015

Future Plans

We expect that our future plans will continually evolve based on user feedback, research priorities, and the dynamic nature of the Google Cloud Platform. Tell us what is important to you at feedback@isb-cgc.org

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5 ISB-CGC Web Interface

The documentation contained in this section is for version 1.0 of the ISB-CGC web interface.

Over time we will be updating and enhancing this web interface based on your feedback. We welcome your ideas and needs. Please use this link to provide them.

1.5.1 Overview

The ISB-CGC web application functionality is accessed through a Google account identity (freely available with a new account or by linking to an existing email account). If you have not logged into the ISB-CGC application you will be presented with this page:
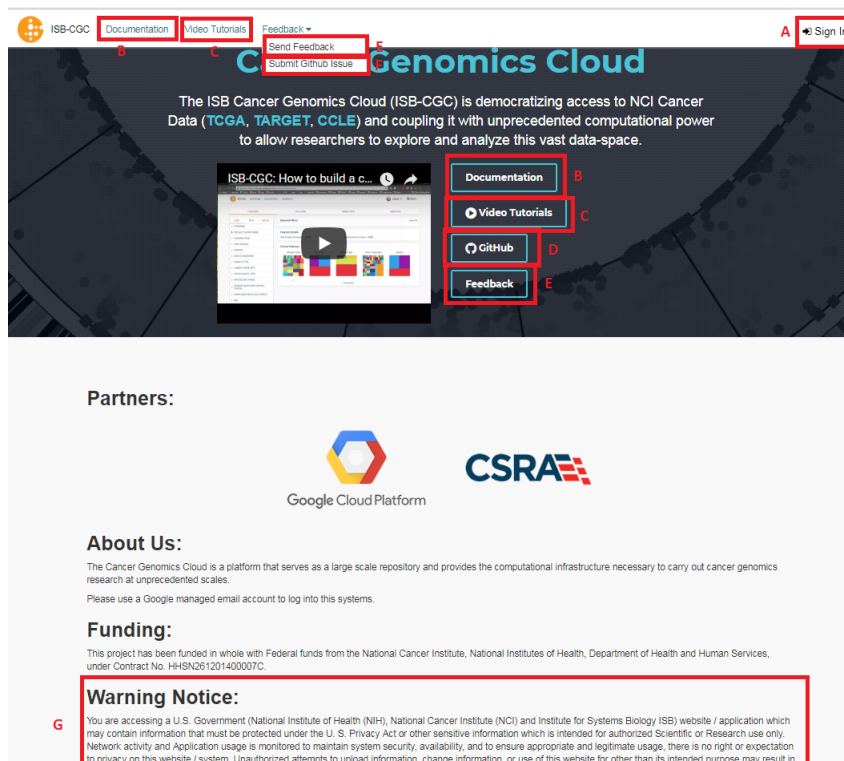
You login through the "Sign In" link at the upper right of the initial page (label "A" in the image above). Also on this page are links to project documentation (B), our public video tutorials (C), our public GitHub repository (D), a link to provide feedback (E), a link to submit an issue to Github (F), and our required "Warning Notice" (G) indicating that this is a U.S. Government Sponsored Website and by using it you are accepting the policies associated with its use.

Upon signing in with a Google account identity, you will be presented with the following page:

This is your personal "Dashboard" where your Analyses, Gene and miRNA Lists, Variable Lists, Cohorts, and Saved Programs are readily accessible. Additional documentation describing how to use each component of this user interface are provided in the individual subsections of this documentation.

Multiple Sample Analyses can be grouped into Workbooks (and saved for later use, editing, and sharing). Workbooks are used to group together multiple related analyses, and can be used for sharing groups of analysis results with specific groups of people. For example, you may use one Workbook for an on-going study of gene mutations and pathways involved in Head and Neck Cancer (with one research group you are part of), and use a different Workbook for another on-going study with a different set of collaborators in which you are investigating survival-time after diagnosis for patients with different types of lung cancers. Think of workbooks as containers in which you can create and group related analyses, and which you can share with specific colleagues.

IF YOUR SCREEN LOOKS LIKE THIS (how do I get to the main screen?): If your screen looks like the image below (I am logged in (A in image) but I can't see my analyses pages ("Your Dashboard" - image above)) that is because some browsers save your Google Login as a cookie, and automatically sign you in. To get to "Your Dashboard" click on the "ISB-CGC" icon in the upper left (B in the image). This will always take you to the main analysis dashboard screen.
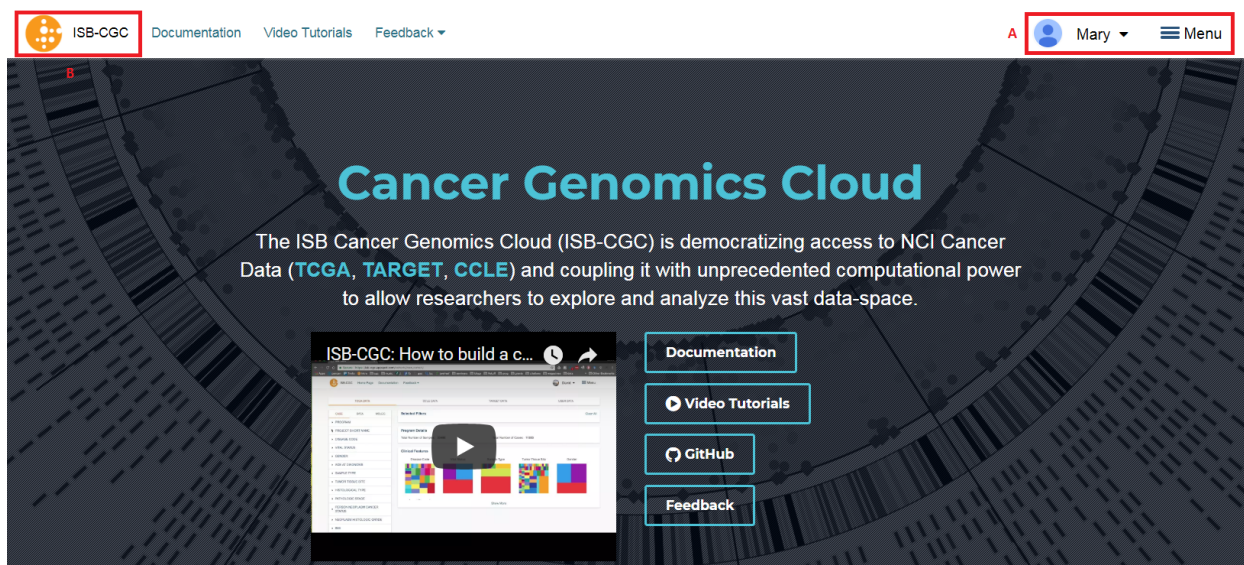
Breadcrumbs show you where you are in the Web Interface as you move from one section to another (figure below). These are live links, and can be used to rapidly navigate from one section of the interface to another.

The data that is being manipulated with the Web Application is the same data that is available through the programmatic APIs. Details describing how to access these data are provided in detail in specific documentation sections elsewhere in the documentation.

The Web Application was optimized for use with the Google Chrome web browser. Most of the functionality should work with recent versions of other web browsers (e.g. Firefox, Safari, Internet Explorer). If you find an issue and you are not using Chrome, please try using Chrome to see if the issue appears to be browser-specific.

> Also please note the system is set in Pacific time, so if you see some inconsistencies with the time in the workbooks or cohorts you generated in the last updated section it could be due to this fact.

If you encounter issues or have questions, please use our feedback forum.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.2 Accessing Controlled Data

Accesssing **controlled data** is done in two different manners, depending if you are doing it through interactive computing (*eg* the Web App or R Studio), or programmatically (*eg* a program running from a Google Virtual Machine Compute Engine you have started). In some cases you will be using your *personal* credentials while in other cases a "service account" will be acting on your behalf, using its own credentials. Below the methods are described. Please note, you can use both methods at the same time, they are not mutually exclusive.

Interactive Access to Controlled Data

Before you can access *any* controlled-data hosted by the ISB-CGC, you must first associate (or "link") your Google identity (which you use to sign in to the ISB-CGC Web App and access the Google Cloud) with a valid NIH login associated with a dbGaP data-access request (either an eRA account ID or an NIH account User ID). This is done through the Web App: you will first be redirected to an NIH login page, and once you have successfully authenticated, ISB-CGC will store an association between your NIH identity and your Google identity. (Note that this should be a one-to-one association.)

Once you have authenticated, ISB-CGC will check which dataset(s) e.g TCGA controlled data and/or TARGET controlled data you have been authorized (by dbGaP) to access. ISB-CGC obtains an updated whitelist for each of the hosted datasets from dbGaP every day. If you have just recently been granted access by dbGaP, there may be a 24 hour delay before you will be able to request access to these data on ISB-CGC.
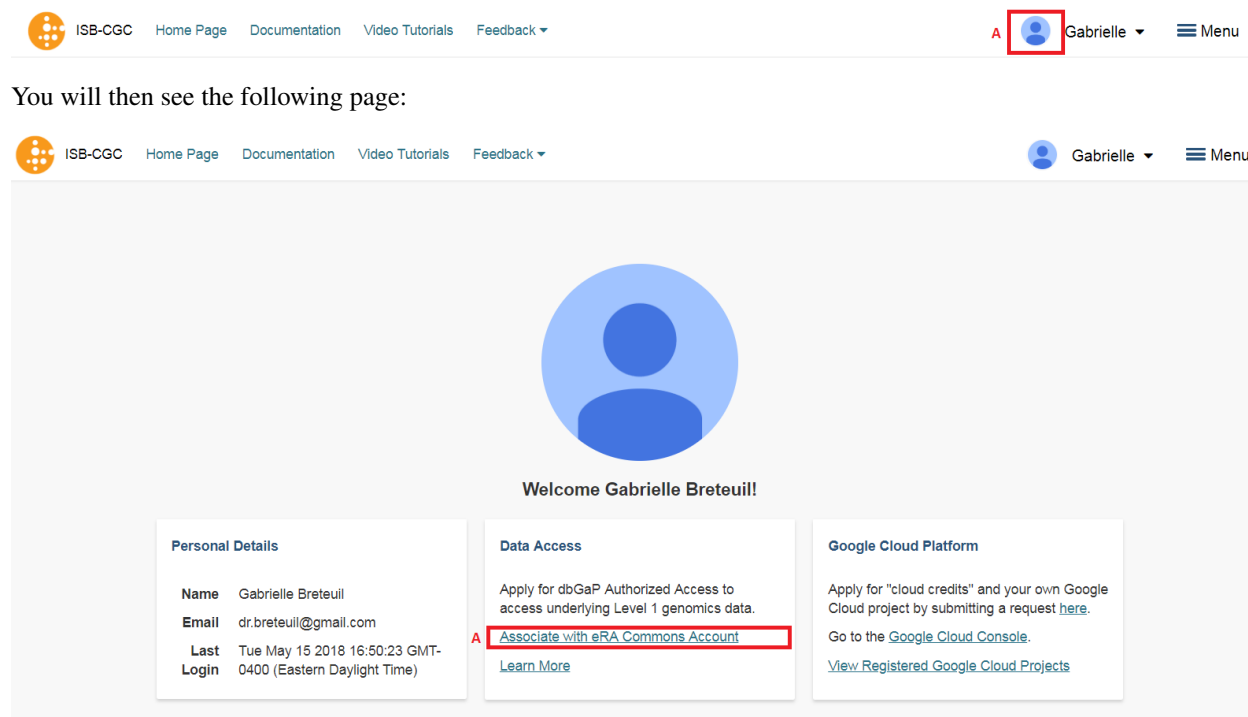
Visit electronic Research Administration (eRA) for more information on registering for a NIH eRA account. NIH staff may utilize their NIH log-in. (For additional instructions, please refer to Data Access Request Instructions, dbGap Data Access Request Portal, and Understanding Data Security). Please be sure to review the Data Use Certification Agreement for TCGA controlled data and TARGET controlled data.

Once you have authenticated to NIH via the Web App, and your dbGaP authorization has been verified, the Google identity associated with your account will have access to the controlled-data for 24 hours.

For more information on applying for dbGaP authorization to access controlled data, please see our Frequently Asked Questions (FAQ) page or the "How to" Apply for Controlled Access Data Video.

Linking your NIH and Google identities

To link your NIH identity with your Google identity (ie the Google account you used to login to the ISB-CGC system), select the "persona" icon next to your login name (A in the image below) after you have signed in to the ISB-CGC Web App.

You will then see the following page:

You will see a pop up describing all the steps needed to link you NIH Identity to the Data Commons Framework(DCF).
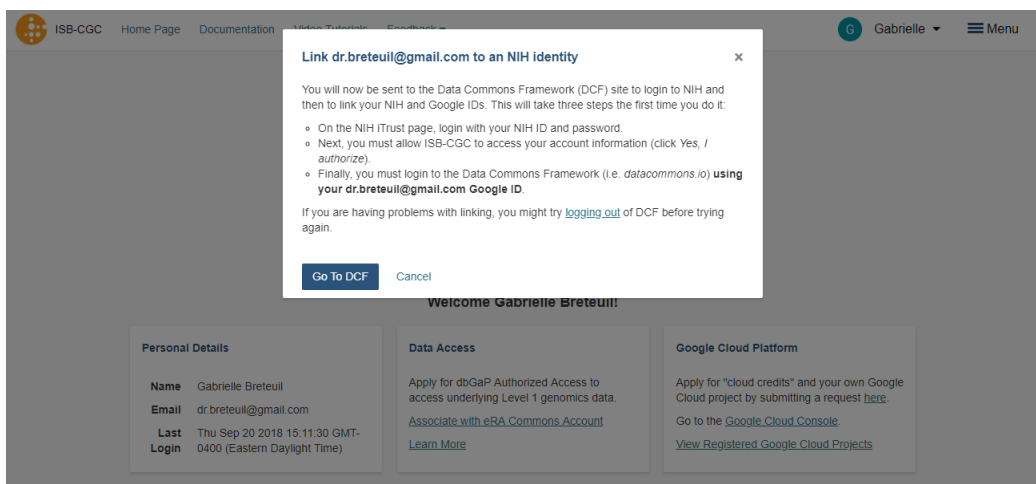
Now you need to associate your Google identity with your NIH identity. (Your NIH identity is the one associated with your dbGaP application and authorization to work with controlled data.) To do this, select the "Associate with eRA Commons Account" link (highlighted in diagram above, and labeled A). You will then be re-directed to an NIH login page to be authenticated by NIH:

If you have an eRA identification, use this to sign in through panel A (see example above). If you have an NIH PIV card, use that to sign in through panel B on this page (see above). Once you have been authenticated by NIH, and your NIH identity has been verified to be on the current dbGaP whitelist, you will have access to controlled data for 24 hours.
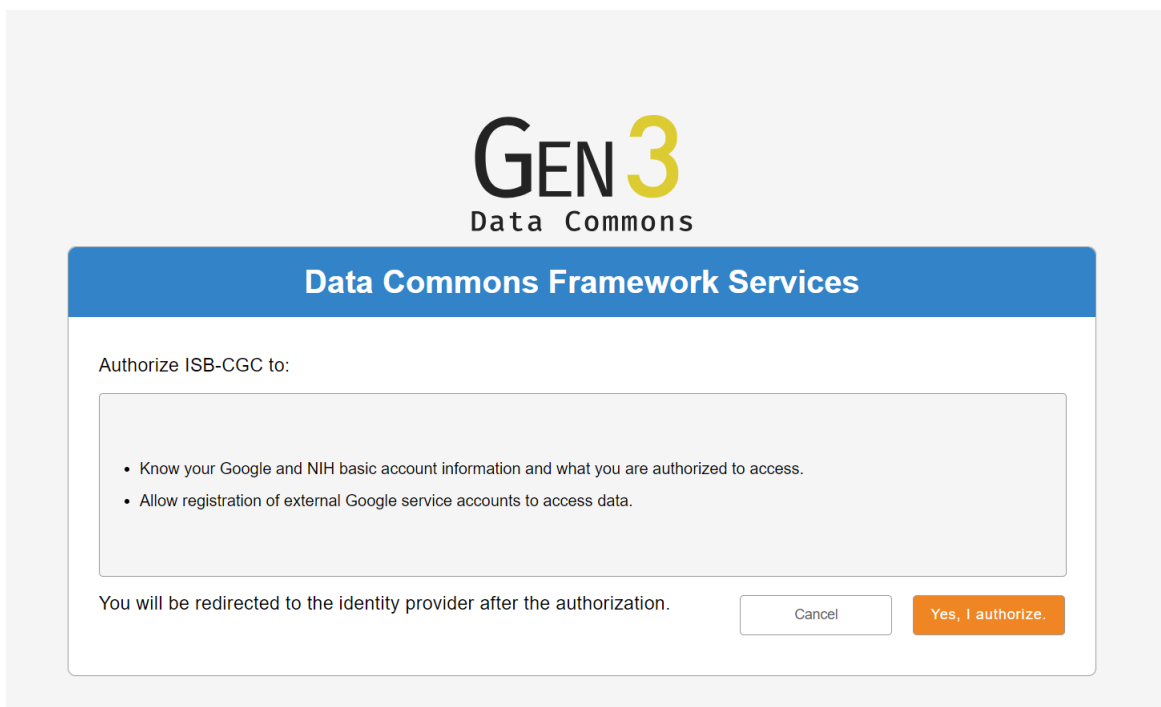
Select the Yes, I Authorize button at the bottom right of the page to authorize the Data Commons Framework to authorize your Google identity with controlled data.
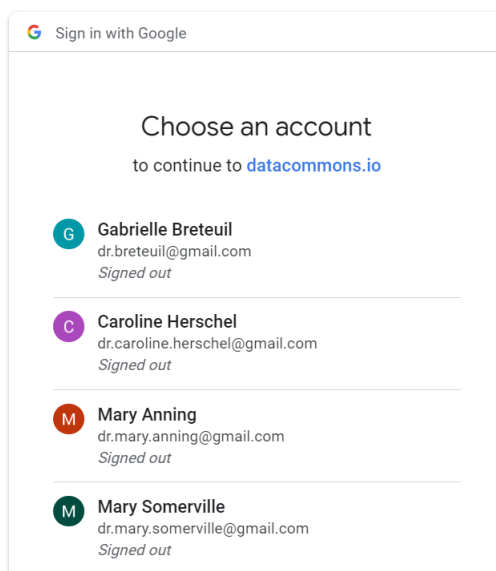
Select the email you used to originally log into the ISB-CGC web application to finalize the authorization.

Once logged in through eRA identification you are re-directed to the user details page and given Warning Notice referring to abiding by the rules and regulations provided by the DUCA Use Agreement. Please refer to image below.

Please note: the ISB-CGC system will enforce a one-to-one relationship between NIH identities and Google identites. In other words, a single NIH identity may not be used to attempt to gain access to controlled data by multiple, different Google identities. If you need to *unlink* your eRA account from your Google account (for example if you want to change which Google identity you use to sign in to the ISB-CGC platform), you may do so by selecting "Unlink <GoogleID> from the NIH username <eRA Commons ID>" (link B in the screen above).

In the unusual instance that your NIH identity has been registered with another Google identity (*eg* with another Google identity you own), you will see the screen below:

If this happens, please sign in with that other account and "unlink" your eRA from that account i (see description above). You will then be able to register your eRA account with the desired Google identity. If you are not able to resolve the issue, contact us at feedback@isb-cgc.org and we will help you resolve it.

To end your Web App session, just "Sign Out" by using the pull-down below your name (see image below, A). After you sign out from the ISB-CGC Web App, your Google identity may still be signed in to your browser, so you may want to also sign out of the browser.

Extending Your Access by 24 hours

Once you have received permission to view controlled access data, your user login page will look like the screenshot below. If you need to extend your access to controlled data for another 24 hours from now (*eg* if you have a compute job which is using these Google credentials to access controlled data and it is still running), select the link "Extend controlled access period to 24 hours from now" (red box on figure below). Your time of access will be extended to 24 hours from the time you push the link.
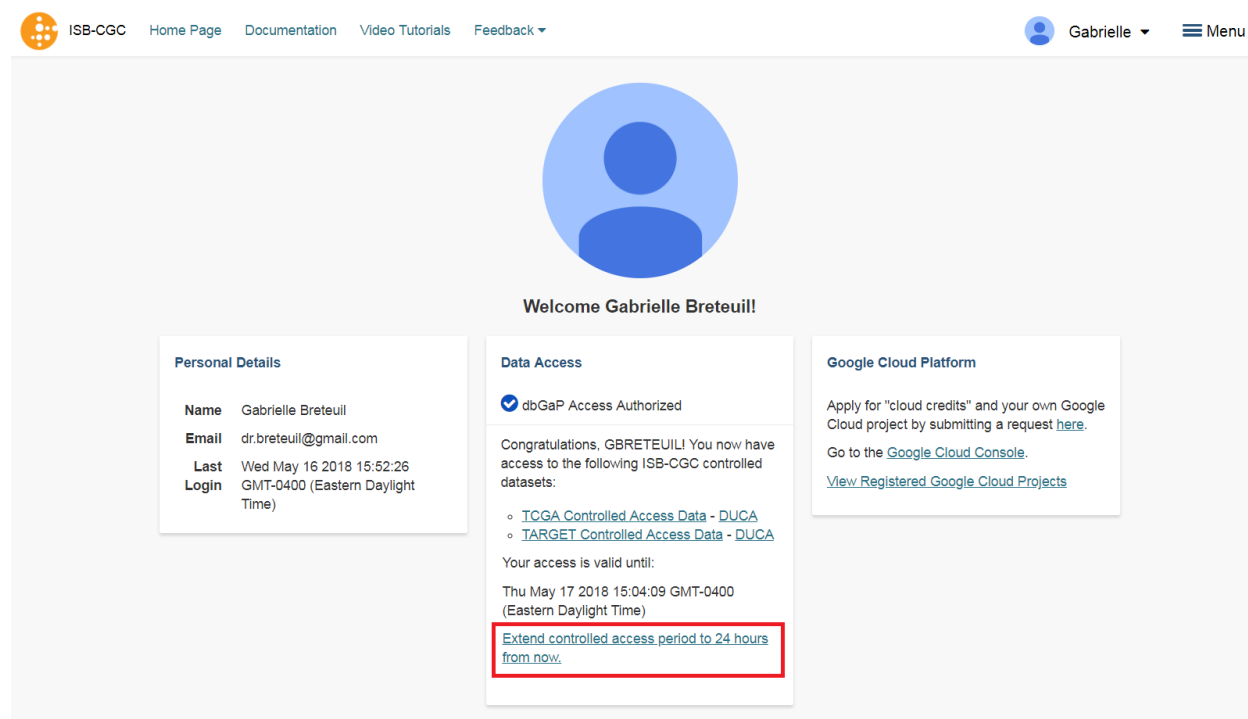
Accessing Controlled Data from a GCE VM

This section only applies to ISB-CGC users with access to a Google Cloud Platform (GCP) project. GCP projects are automatically configured with a "Compute Engine default service account" which you can find on the IAM & Admin page of the Cloud Console. You can create additional service accounts for special purposes, but most users will be able to just use this one default service account.

When running on a Google Compute Engine (GCE) VM (virtual machine), a "service account" associated with your Google Cloud Project (GCP) is generally acting on your behalf and those are the credentials being used rather than your personal credentials. (If you want to learn more about service accounts, please refer to the Google documentation.)

In order for this **service account** to access controlled data, you must register it with ISB-CGC. Once this process has completed successfully, this service account will be able to access controlled data for up to 7 days.

NOTES:

- to allow flexibility while working with different research teams and different processes, you can have many GCPs registered with ISB-CGC, as well as many service accounts registered per GCP

- if the service account (*ie* any program running on a VM using the service account's credentials) tries to access controlled data after the 7 day expiration, it will get an Access Denied error; to prevent this from causing problems with long-running jobs, you can extend access by another 7 days (see below);

Requirements for Registering a Google Cloud Project Service Account

To be able to register your GCP Project and at least one service account to access controlled data the following must all be true:

- You must be an **owner** of the GCP project (because you will need to add an ISB-CGC service account as a new project member and a DCF service account as a new project member)

- 907668440978-oskt05du3ao083cke14641u35deokgjj@developer.gserviceaccount.com

- fence-service@dcf-prod.iam.gserviceaccount.com

- At any time, ALL members of the project MUST be authorized to use the data set (*ie* be a registered dbGaP "PI" or "downloader") (see dbGap Data Access Request Portal, and Understanding Data Security for more details).

- All members of the project have signed in to the ISB-CGC Web App *at least once*

- All members of the project have authenticated via the NIH login page and thereby linked their NIH identity to their Google identity

- The GCP project can not be associated with an Organization

- No Google Groups or other multi-member identifiers (e.g. all authenticated Google users) have been provided with a project role

- The GCP project must have the ISB-CGC monitoring service account (SA) assigned to an Editor role

- All SAs with roles in the project must belong to the project, with the exception of the ISB-CGC monitoring SA; this means that all Google-managed SAs with project roles must belong to the project as well

- The SA you are registering cannot be the ISB-CGC monitoring SA, or SAs from other projects

- You have not created any keys for any SAs in the project

- No IDs have been assigned roles on any SAs in the project

If ANY of these requirements are not met, your GCP and ANY associated service accounts will **not** be able to access controlled data. An automated email will be sent to the GCP project owner(s) if data access is revoked.

Registering your Google Cloud Project Service Account

To register your GCP and its Service Account with ISB-CGC, select the "persona" icon next to your login name (see first image above), which takes you to the following page:
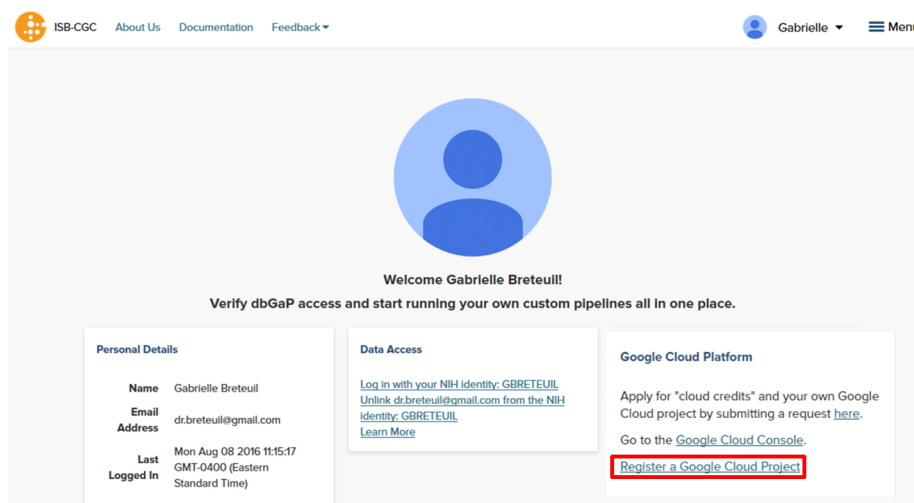
Select the "Register a Google Cloud Project" link. That takes you to the following page:
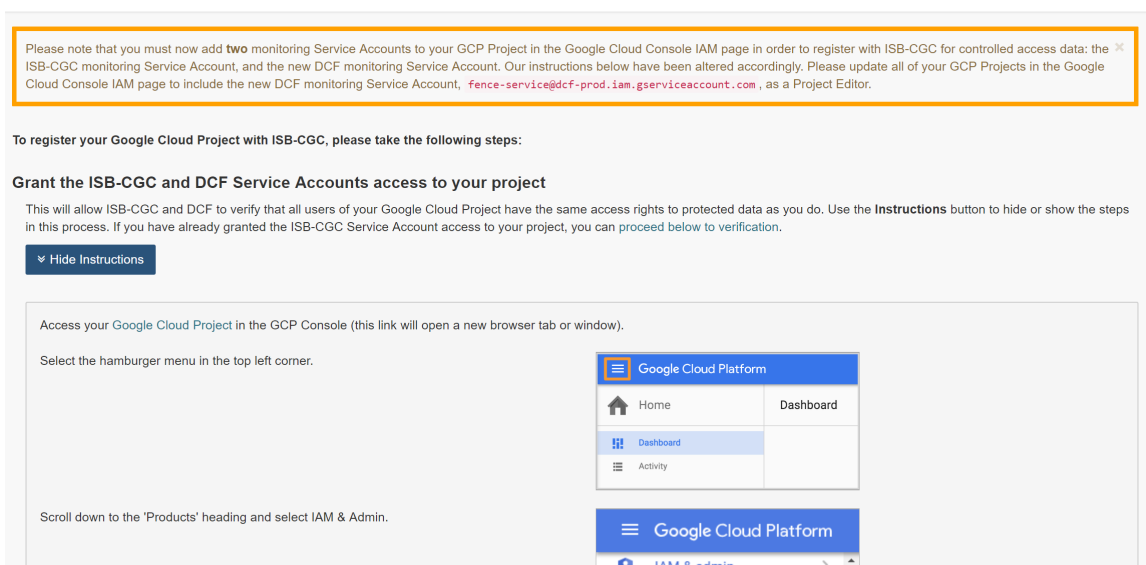
Please fill out the form following the instructions that are provided. You can "hide" the instructions by selecting the blue Instructions button. You must enter your GCP ID and enable the isb-cgc service account as an editor in your project to move on to the next step.
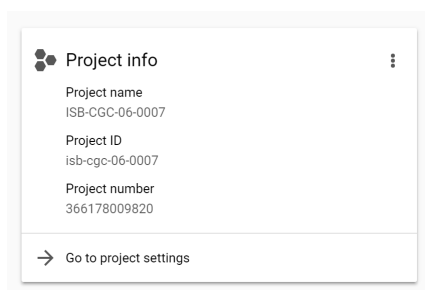
Please be sure to add both service accounts listed below. If you don't add both service accounts you will run into issues viewing the controlled data in ISB-CGC.
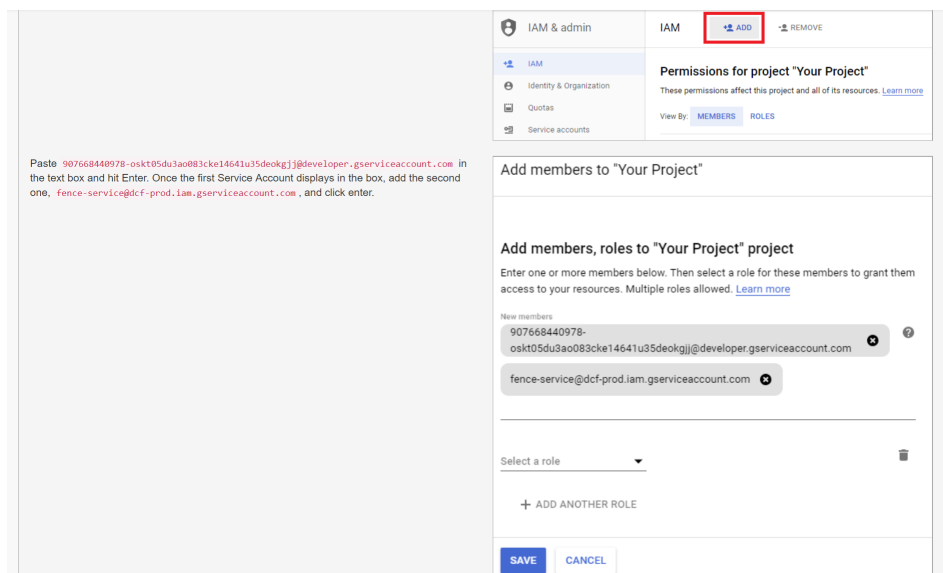
Once you have completed these steps you will be presented at the bottom of the same page a listing of the members of your GCP you registering (see screenshot below):
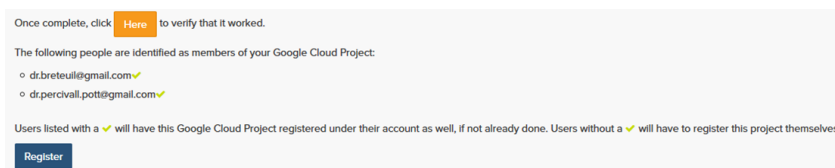
Pushing the "Register" button will take you to the next screen:
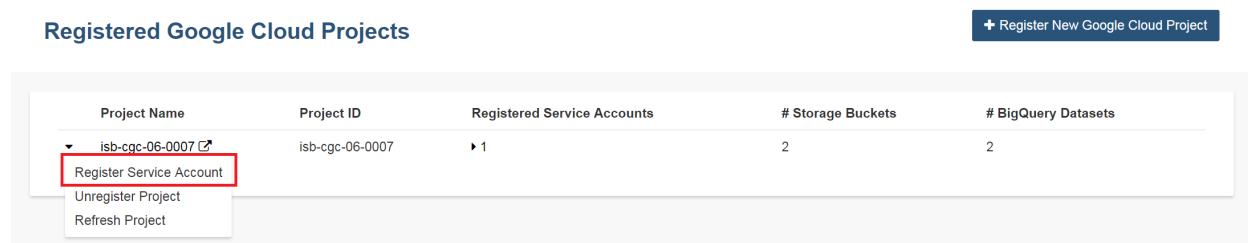
Select "Register Service Account" from the drop down menu on the left of the GCP you want to add a service account to. By default, there will be the Compute Engine Default service account in the Enter the service account ID text box (see screenshot below). Addtionally, select the programs you wish to gain access to by selecting the checkbox to the associated Controlled Dataset(s) you plan to access. Currently you can select either Controlled TCGA data or controlled TARGET data to gain access to.

If you recieve the error message listed below, this signifies you need to enable the Default Compute Engine API for your Google Cloud Project. For more information on how to enable all the API's you will need to work on a Google Cloud Project please go here.

Once you click the "Verify Service Account Users" at the bottom of the page, you will be presented with multiple lists. You will be presented with the Verification Results, Google Cloud Project User ISB-CGC Registration and Identity Linkages, Dataset Permissions Verification, Registered Service Account Verification Results, Google Cloud Project Verification Results, and the Google Cloud Project Service Account Verification Results (see screenshots below). All columns MUST have a green check-marks in them for each user before your service account can be registered.

Registered Service Account Verification Results:

✓ The requested service account "██████████████@developer.gserviceaccount.com" meets all requirements.

Google Cloud Project Verification Results:

✓ Google Cloud Project ████████████ meets all requirements
✓ All Google Cloud Project roles meet requirements. Roles are assigned to either service accounts or to users who have registered with the Data Commons Framework.

Google Cloud Project Service Account Verification Results:

ID	Status	Message
██████████@genomics-api.google.com.iam.gserviceaccount.com	✓	
██████████@containerregistry.iam.gserviceaccount.com	✓	
██████████@gcp-sa-websecurityscanner.iam.gserviceaccount.com	✓	
██████████.iam.gserviceaccount.com	✓	
██████████@cloudservices.gserviceaccount.com	✓	
██████████.iam.gserviceaccount.com	✓	
██████████@dataflow-service-producer-prod.iam.gserviceaccount.com	✓	
██████████@compute-system.iam.gserviceaccount.com	✓	
██████████.iam.gserviceaccount.com	✓	
██████████.iam.gserviceaccount.com	✓	
██████████@container-engine-robot.iam.gserviceaccount.com	✓	

We have verified that your Google Cloud Project meets all requirements, and that all of the users in that project have permission to access the proposed datasets.

[Register]

If all the requirements for registering a service account are met, the account will be registered. If not, the service account will only be registered for Open Datasets. The final screen below shows the final registered data set (shown by selecting the drop-down menu beside the service account count highlighted in red).

▼ ISB-CGC-06-0007 ⧉	isb-cgc-06-0007	▼ 1	0	1

Service Account	Authorized Controlled-Access Datasets	Current Access Expires	
366178009820-compute@developer.gserviceaccount.com	TCGA Controlled Access Data, TARGET Controlled Access Data	Dec 13, 2018, 12:25 p.m.	↻ 🗑 ➕

Managing your Google Cloud Project(s) and Service Account(s)

Once your GCP(s) and Service Account(s) are registered, you can add or remove additional service accounts by following the instructions below. You can also extend the use of a service account for another 7 days, or reauthorize a service account after you have corrected errors that previously caused it to have its permissions revoked.

Adding additional Google Cloud Projects

To add additional Google Cloud Projects (GCPs) that you own to be able run programs programmatically select the "+ Register New Google Cloud Project" button from the "Registered Google Cloud Projects" page (see screenshot below).

Registered Google Cloud Projects   [+ Register New Google Cloud Project]

	Project Name	Project ID	Registered Service Accounts	# Storage Buckets	# BigQuery Datasets
▼	isb-cgc-06-0007 ⧉	isb-cgc-06-0007	▶ 1	1	1

Deleting Google Cloud Projects

To delete a GCP that is registed, select the "Unregister Project" button from the dropdown menu beside the project your are removing on the "Registered Google Cloud Projects" page (see screenshot below).

Registered Google Cloud Projects   + Register New Google Cloud Project

Project Name	Project ID	Registered Service Accounts	# Storage Buckets	# BigQuery Datasets
▼ isb-cgc-06-0007 ⬈	isb-cgc-06-0007	▸ 1	2	2
Register Service Account				
Unregister Project				
Refresh Project				

Adding additional service accounts to a given Google Cloud Project

To add additional service accounts to a given GCP reselect the "Register Service Account" from the dropdown menu beside the project that has the service account (see screenshot below).

Registered Google Cloud Projects   + Register New Google Cloud Project

Project Name	Project ID	Registered Service Accounts	# Storage Buckets	# BigQuery Datasets
▼ isb-cgc-06-0007 ⬈	isb-cgc-06-0007	▸ 1	2	2
Register Service Account				
Unregister Project				
Refresh Project				

Adjusting a Service Accounts using the Adjust Service Account page

To add or remove a controlled dataset from one specific service account from this feature. If you select the plus "+" sign icon next to the trash can (see screenshot below).

Project Name	Project ID	Registered Service Accounts	# Storage Buckets	# BigQuery Datasets
▼ isb-cgc-06-0007 ⬈	isb-cgc-06-0007	▼ 1	2	2
Service Account		**Authorized Datasets**		**Date Last Activated**
366178009820-compute@developer.gserviceaccount.com		All Open Datasets, TCGA Controlled Access Data, TARGET Controlled Access Data		Nov 30, 2017, 12:38 p.m. ↻ 🗑 ⊞

Deleting Service Accounts from Google Cloud Projects

To delete a service account from an GCP (not allowing it to be used to programmatically access controlled data), push the "trashcan" icon beside the service account (see screenshot below).

Project Name	Project ID	Registered Service Accounts	# Storage Buckets	# BigQuery Datasets
▼ isb-cgc-06-0007 ⬈	isb-cgc-06-0007	▼ 1	2	2
Service Account		**Authorized Datasets**		**Date Last Activated**
366178009820-compute@developer.gserviceaccount.com		All Open Datasets, TCGA Controlled Access Data, TARGET Controlled Access Data		Nov 30, 2017, 12:38 p.m. ↻ 🗑 ⊞

Extending Your Service Account Access by 7 Days

Once you have registered a Service Account, you have 7 days before the access is automatically revoked. To extend the service account access another 7 days (*eg* if your program is still running), select the "refresh" icon beside the service account (see screenshot below).

Reauthorizing a Google Cloud Project(s) Service Account(s)

Your service account may have its permissions revoked (because, for example, the 7-day limit has expired, or you have added a member to the GCP who is not authorized to use that controlled data). If permissions were revoked because an unauthorized user was added to the project, the Google Cloud Project owner will be sent an email specifying the Service Account, GCP Project, and user which resulted in the access being revoked. To reauthorize the service account 1) remedy the problem that resulted in access being denied, and 2) select the "adjust" icon beside the service account (see screenshot below) and add the controlled datasets to the service account.

Google Cloud Project Associated to an Organization Will NOT Work with controlled data

If your Google Cloud Project is associated to an organization you will be unable to register the service account to controlled data. You will return an error message similar to this saying, "GCP cgc-08-0126 was found to be in organization ID 8784632854871; its service accounts cannot be registered for use with controlled data." This is mainly to due with the fact ISB-CGC cannot see the permissions assocaited to the organization project therefore is a security risk. We are currently working with Google to resolve this issue.

Your Responsibilities

You should think about securing controlled data within the context of your GCP project in the same way that you would think about securing controlled data that you might download to a file-server or compute-cluster at your own institution. Your responsibilities for data protection are the same in a cloud environment. For more information, please refer to NIH Security Best Practices for Controlled-Access Data.

NIH has tried to provide as much information as possible for PIs, institutional signing officials (SOs) and the IT staff who will be supporting these projects, to make sure they understand their responsibilities." (Ref: The Cloud, dbGaP and the NIH blog post 03.27.2015)

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.3 Menu Bar

Clicking on the Menu icon or word in the upper-right corner of your browser window (next to your name), will insert the blue menu bar in your current view. You can make this menu bar disappear by clicking the on the blue **X** or the word Menu again.

The **MENU** bar is a graphical control element which contains application navigational drop downs (sub-links). The Menu bar's purpose is to supply quick and common links for application-specific functions / features such as:

- **DASHBOARD** - This link takes you to *Your Dashboard* main page.
- **WORKBOOKS - Workbooks store the Analyses you create – and their related data. You can create worksheets as you exp**

  - *Recent* - Displays your created workbooks and allow quick navigation to them;
  - *Saved* - Displays all your saved workbooks and allows you to edit, duplicate or delete the workbook;
  - *Create a New Workbook* - Quick Link to Workbook Creation, where you can select the data source

- **PROGRAMS - These are shortcuts to the programs you have created if you uploaded your own data.**
  - *Saved Programs* - Here you Can:
    * Edit or delete a Saved Program

* Start a New Workbook

* Create a New Program

   – *Upload Program Data* - Here you can:

      * Create a new program for analysis. To create a new program you provide a name for program, name for your project, and attach files that meet our Data Type requirements. Please see Program Data Upload for more information on data type accepted by the ISB-CGC.

      * Save or cancel a new creation

   – *Public Programs* - Here you can:

      * View the programs and project that are currently in the ISB-CGC system.

• **ANALYSES - From here you can Create, Edit Details, Duplicate, Delete, or Share Analyses. You can use a specific analysi**

   – *Bar Chart*

   – *Histogram*

   – *Scatter Plot*

   – *Violin Plot*

   – *Cubby Hole Plot*

   – *SeqPeek*

   – *Browse All Analyses*

• **GENES & miRNA** - From this Menu selection you can **Manage Gene and miRNA Favorites**, **Create Gene and miRNA Favorite(s)** or **Select Genes and miRNAs for a New Workbook**.

   For additional details, see Read the Docs - Gene & miRNA Favorites. Each of these categories provide a quick link to additional application-specific functions / features such as:

   – *Manage Gene & miRNA Favorites* - Here you can:

      * Edit or Delete a Saved Gene and miRNA Favorite(s)

      * Start a New Workbook

      * Create a New Gene and miRNA Favorites

   – *Create Gene & miRNA Favorite* - Here you can:

      * Create a Gene & miRNA Favorite for Analysis. To Create a New Gene and miRNA Favorite - You provide a name and select the Gene and/or miRNA. You can upload a stored Gene and miRNA List or type in Gene name or miRNA (**Note:** This will auto fill as you type in Gene name or miRNA name). To aid in Gene selection, you can access the HGNC portal (Hugo Gene Nomenclature Committee) via the "**View Gene Identifiers**" link under this Menu selection. Also, to aid in miRNA selection, you can access the miRBASE via the "**View miRNA Identifier**" link next to the View Gene Identifiers link.

      * Save or Cancel a new creation.

   – *Select Genes & miRNA for a New Workbook* - This sub-menu has two features:

      * Apply to New Analysis - Select a Favorite(s) Gene and miRNA from the list shown of stored Favorites to Analyze

      * Add (+) Apply to New Analysis - Basically navigates back to the **Create Gene and miRNA Favorite** (See description above)

- **VARIABLES - This sub-menu allows you to Manage Variables Favorite or Create New Favorite (see descriptions below) F**

  - *Manage Favorite Variable(s) Lists* - Shows your saved Variables as Favorites:

    * Edit

    * Delete

    * Start New Workbook - (Create a New Workbook using the selected Favorite Variables)

  - *Create Favorite Variable(s) List* - Here you "Name" your new favorite and select variables from four (4) available data sources to incorporate in your analysis-

    * Common Variables

    * Favorite(s) Saved

    * Programs (Previously Uploaded and Saved)

  - *Select Variables for a New Workbook* - This sub-menu has two features:

    * Apply to New Worksheet - Select a Favorite(s) variables from the list shown of stored Favorites to Analyze

    * Add (+) Apply New Variable List - Basically navigates back to the **Create Variables Favorite** (See description above)

- **COHORTS** - Here you can **Manage Saved Cohorts**, select **Public Cohorts** and **Select Cohorts for a New Workbook** or **Create your First Cohort** if it's empty. For additional details, see Read the Docs - Cohort Favorites.

  - *Manage Saved Cohorts* - There are two tabs here

    * **Saved Cohorts - Displays previously created cohorts which can be selected. If not cohorts exist, you can create yo**

      · Create a "New Workbook" from a saved Cohort

      · Delete a Saved Cohort

      · Set Operations (i.e., Union, Intersection or complement) from a Base or Subtracted Cohort.

    * **Public Cohorts - Displays any public cohorts which can be selected.**

      · Create a "New Workbook" from a saved Public Cohort

      · Set Operations (i.e., Union, Intersection or complement) from a Base or Subtracted Cohort.

  - *Public Cohorts* - This is a quick link performing the same functions described in the respective tabs of Saved Cohorts above.

  - *Select Cohorts for a New Workbook* - This is a quick link performing the same functions described in the respective tabs of Saved Cohorts and Public Cohorts above.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.4 Workbooks

Workbooks store the Analyses you create – and their related data. Basically, the worksheets you create to conduct analysis based

- Group together multiple related analyses,

- Share analysis results with specific groups of people,

For example, you can create a Workbook (i.e., Disease A) which consists of identifying gene mutations and pathways involved in Head and Neck Cancer (and share it with research Group A).

And to create another Workbook (i.e., Disease B) with a different group of researchers (Group B) investigating the average time after diagnosis of death for different lung cancers. Think of workbooks as virtual "excel spreadsheets" that various related analyses can be created in individual Worksheets ("Tabs" within the spreadsheet) and grouped together in one Workbook (the overall spreadsheet).

Additionally, you can:

- Save a Workbook for later use, or

- Edit an existing Workbook.

Creating and Saving a Workbook

From your Dashboard, you will find the "Saved Workbooks" panel. This panel displays any previously created, Saved Workbooks, also you are able to "Create a New Workbook". If you do not have any saved workbooks you will see "Workbooks store the analyses you create, and their related data." text in the panel.

To create a **Workbook** from Your Dashboard, click on the **"Create a New Workbook"** link in the "Saved Workbooks" panel. This will take you to the Workbook creation page.

Note: If you wish to use your own data in graphing, please review the documentations on how to upload your own data and on how to graph your own data. Using your own data uses a slightly different approach than is described here.

ANNOUNCEMENT: For OncoPrint, OncoGrid, and SeqPeek analyses we now provide a default gene list. Genes with consensus score of 6 or higher are added to the default gene list. (Ref: Bailey et al., Cell. 2018 Apr 5;173(2):371-385.e18. doi: 10.1016/j.cell.2018.02.06)

1. From the Workbook creation panel you will first select one of six analysis types (i.e., Bar chart, Histogram, Scatter Plot, Violin Plot, Cubby Hole Plot or SeqPeek).

   - **Analysis Type Description**

     - **Bar Chart** - This chart is used to plot a single categorical feature for one or more cohorts. It generates vertical lines to represent the type of data being used. The X axis shows categorical information being used while the other y axis, displays categorical data chosen in the edit analysis settings.

     - **Histogram** - This chart is used to plot a single numerical feature for one or more cohorts. It generates vertical lines to represent the type of data being used. The X axis shows numerical information being used while the other y axis, displays numerical data chosen in the edit analysis settings.

     - **Scatter Plot** - This chart is used to plot two numerical features (x & y axis) for one or more cohorts. Can also color code points by a single categorical feature.

     - **Violin Plot** - This chart is used to plot a categorical feature on the x-axis versus a numerical feature on the y-axis. Points in the plot can be colored by another categorical feature.

     - **OncoPrint** - This chart is used to plot multiple genomic alteration (somatic mutation) events across a set of samples using color-coded glyphs. OncoPrint is developed and provided by cBioPortal.

     - **OncoGrid** - This chart is used to visualize the top mutated genes across programs/projects and the number of cases affected. We are also able to view the mutation frequency, clinical data, data format types, number of gene sets and the number of cases affected.

- **Cubby Hole Plot** - This chart is used to plot two categorical features. Boxes are colored by their related p-values.

- **SeqPeek** - This visualization shows where somatic mutations have been observed on a linear representation of a specific protein. Each horizontal strip represents the protein, with data from different tumor types (aka cohorts or studies) shown stacked one on top of the other.

A researcher now has the option to make the axis logarithmic if the plot can display continuous numerical data for eg. mRNA expression levels.

Note: For Violin Plot and Scatter Plot you can select multiple cohorts as your Color By Feature. This will cause the Legend to list all the cohorts that the sample is associated to. Please be aware you'll end up with lots of permutations if you have lots of samples that belong to many different cohorts.

2. You will then select **Genes and miRNAs or Variables** (or, optionally both)

- **Genes and miRNAs** - This will display previously created "Gene and miRNA Favorites" which can be "Applied to Analysis" (see Gene and miRNA Favorites for details) or you can apply / create a "New Gene and miRNA List").

You begin by naming the data set (Gene Type or Project Specific); the Gene and miRNA list you create here will automatically be added to your Gene and miRNA Favorites list and can be selected for additional analysis later.

Next select the Gene(s) and miRNA(s) you're interested in (you can upload a specific list you've previously created/stored, select new Genes and miRNA by typing them into the input box (you will see that Genes and miRNAs will auto-display). If duplicate listings are entered they will be marked for your deletion or automatically dropped when saved. If an error or unknown item is entered it will also be flagged for your attention.

To aid in Gene selection, you can access the HGNC portal (Hugo Gene Nomenclature Committee) via the "View Gene Identifiers" link under this Menu selection. To aid in miRNA selection You can also access the miRBase portal via the "View miRNA Identifiers" for readily available miRNA identifiers.

This allows you to focus on specific results or queries.

- **Variables** - (this will display previously created "Variable Favorites" which can be "Applied to Analysis" (see Variable Favorites for details) or you can apply / create a "New Variable List").

You begin by naming the new Variable; the Variable you create here will automatically be added to your Variable Favorites list and can be applied to other analysis later.

Next you can select variables from five (5) available data sources to incorporate in your Variables-

- *TCGA* - Common (22 options) and clinical search feature

- *CCLE* - Common (8 options) and clinical search feature

- *TARGET* - Common (16 options) and clinical search feature

- *Favorite(s)* - (Previously created variables which were Saved)

- *User Data* - Every program and data variable you have uploaded into system.

Then **"Apply to Worksheet"**

3. Select your **Cohort** - Cohorts allow the user to create custom groupings of the samples and/or cases that can be used for further analysis.

By clicking 'Cohorts' or the '+' symbol you will be directed to the Cohorts table where the user can either create a new cohort or choose from an existing cohorts. When creating a new cohort, after selecting your criteria you click the "Create Cohort" button. The button will become disabled (dimmed) as the Cohort builds (shown in the image below).
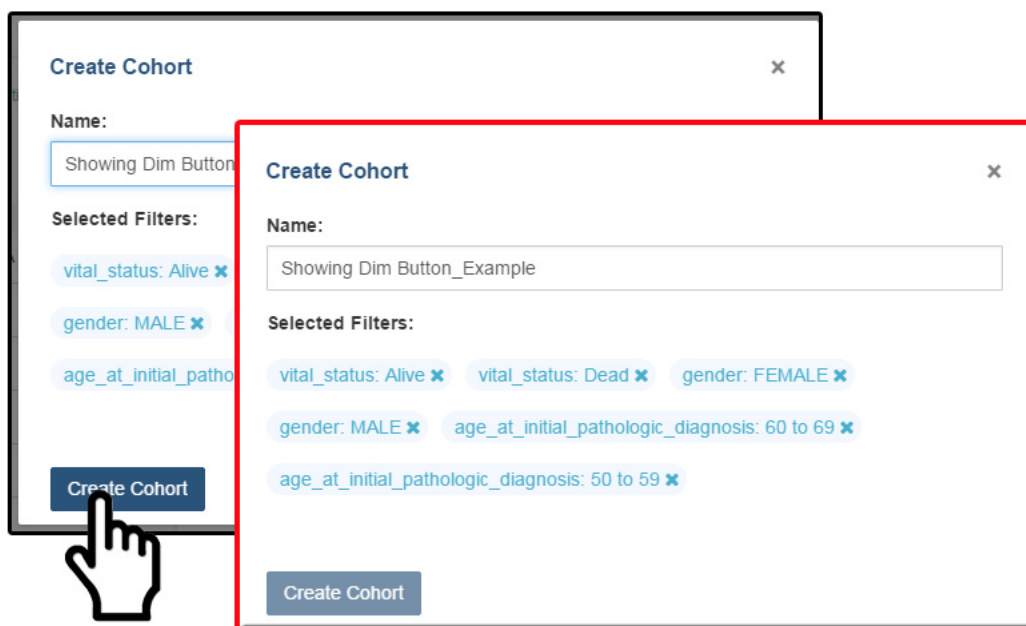
Show Button Dimmed Once Clicked

Once completed you can proceed. The user can also add multiple Cohorts to the worksheet if desired. More information about Cohorts can be found here.

4. Select **Edit Analysis Settings** - This will trigger the Plot setting displaying the applicable x & y axis settings (i.e. Categorical or Numerical based on analysis type selected). Depending on the analysis type selected (e.g., Bar chart, Histogram, Scatter Plot, Violin Plot, Cubby Hole Plot, or SeqPeek) additional specifications may appear for selection.

5. Select **Toggle Sample Selection** - After an analysis option has been selected, using the toggle sample button allows you to create a smaller cohort from within the plot itself.

6. Select **Redraw** - After an analysis option has been selected, using the redraw button will reset the anaylsis to its orginal setting after being zoomed-in, moved place.

7. Select **Download** - After an analysis option has been selected, using the download button will allow you to either download analysis as a SVG, PNG, or a JSON file.

8. Select **Toggle Full Screen** - After an analysis option has been selected, using this button will enable the plot to full-screen.

Sharing Workbooks

This will share the web view of workbooks with users you select by entering the users e-mail. The User will receive an e-mail message with a link to your shared workbook explaining that (you) wanted to share a workbook with (them) and that you have invited them to join. If the email address you entered in not registered in the database you are prompted with a message saying, "The following user emails could not be found; please ask them to log into the site first:(email entered)."

Manipulation of Workbooks

Creating A Worksheet - By selecting the "+" next to an existing worksheet a user can create a new worksheet to create a new analysis. You can give the new worksheet an unique identifier and also give a description for the worksheet. This is ideal by allowing the user to easily have access to different graphs with the same data in the same workbook.

Worksheet Menu - The worksheet will have a section similar to the workbook menu where user can edit, duplicate or delete the worksheet. You can find the worksheet menu bar by clicking the arrow pointing down located next to the name of the worksheet that is opened.

Edit Analysis Settings - This function allows you to select new Plot Setting for selected analysis type.

Please Note: When selecting a gene or miRNA for either the x-axis or y-axis variable you be prompted with selecting a specification. If you select Gene Expression you have the option of choosing a Select Feature. If you select the Copy Number specification you can choose a Value Filter. If you select the Protein specification you can selelct a Protein Filter. If you select the Mutation specification you can select a Value Filter. If you select an miRNA expression you can select a select feature.

Enable Sample section and Edit Analysis Settings - Enable Sample Selection(shown in the image below) allows you to select samples from displayed analysis and save that selection to a new Cohort for further drill down analysis. The Edit Analysis Settings allows you to change the variables you wish to use for your analysis(varies by which analysis you choose). Finally, if you select miRNA you can select specification miRNA Expression, you will be prompted to select a feature.

Analysis Type

-- select an analysis --

◇ Enable Sample Selection
📊 Edit Analysis Settings

❶ To Complete this Analysis:
- You must select an Analysis Type (above)
- You must select Genes or Variables (or, optionally, both)
- You must select a Cohorts

Edit Details - This function allows the user to edit the name of the worksheet and also give a brief description on the worksheet being used for analysis. You can also change the build from HG19 to HG38 using this feature. Changing the build allows to graph data from either builds.

Duplicate - This function allows the user to create a duplicate worksheet in the workbook for further analysis and comparison.

Delete - This function will only appear when you are working with multiple worksheets. This will permanently delete the worksheet from the workbook.

Comments Section

Any user who owns or has had a workbook shared with them can comment on it. To open comments, use the comments button at the top right and select "Comments". A sidebar will appear on the right side and any previously created comments will be shown.

On the bottom of the comments sidebar, you can create a new comment and save it. It should appear at the bottom of the list of comments.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.5 Genes and miRNAs Favorites List

PURPOSE

This feature allows you to create and manage Gene and miRNA lists for use in subsequent analyses. From the Menu - Genes and miRNA selection, you can Manage Gene and miRNA Favorite(s), Create Gene and miRNA Favorite(s) or Select Genes and miRNAs for a New Workbook analysis.

Creating A Gene and miRNA List Favorite

To create a new Gene and miRNA Favorite -

- Begin by naming your new "Favorites list"; you can create many Favorites and use them later when working with workbooks.

- Next specify the Gene(s) and/or miRNA(s) to populate this list: you can upload a pre-existing list, or enter Genes and miRNAs one at a time by typing them into the input box (with auto-completion support).

If duplicate symbols are entered they will be marked for your deletion or automatically dropped when the list is saved. If an unrecognized item is entered it will also be flagged for your attention. To aid in Gene selection, you can access the HGNC portal (Hugo Gene Nomenclature Committee) via the View Gene Identifiers link provided under this Menu selection. To also aid in miRNA selection, you can access the miRBase via the View miRNA Identifiers link provided next to the View Gene Identifier link mentioned above.

Terminology Used for Genes

- The National Human Genome Research Institute (NHGRI) created the Talking Glossary of Genetic Terms to help everyone understand the terms and concepts used in genetic research. In addition to definitions, specialists in the field of genetics share their descriptions of terms, and many terms include images, animation and links to related terms.

Terminology Used for miRNA

- The miRBase created a microRNA database center to enable researchers to understand the published miRNA sequences and annotations.

Resources

A variety of on-line resources exist that may be useful for understanding and working with Gene identifiers, for example:

- Hugo Gene Nomenclature Committee (HGCN) or
- National Center for Biotechnology Information (NCBI)

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.6 Variable Favorites

A variable Favorites list is a way of creating custom groupings of the samples and/or cases that you are interested in analyzing further. For example, you can create a variable favorites list that span across multiple projects, only contain samples for which certain types of data are available, or focus on specific phenotypic characteristics.

Creating and Saving a Variable Favorites List

To create a variable list from the User Dashboard, click on the "Create Variable Favorite" link where you will be directed to the Create Variable Favorite Page.

Variable Favorites List Creation page

Using the provided list of filters on the left hand side, you can select the attributes and features that you are interested in.

By clicking on a program, the field will expand and provide you with additional filtering options in the Data Types section. For example, when to select the TCGA tab you see a common filters section. The common filters section is shared across programs, so if the common filter is selected in one program it will be selected for all programs.

Variable Favorites Filter

Common Filter List by Program

TCGA Common Filter List	CCLE Common Filter List	TARGET Common Filter List
Year of Diagnosis	Gender	WBC at Diagnosis
Residual Tumor	Disease Code	Year of Diagnosis
Neoplasm Histologic Grade	Sample Type	Event Free Survival
Disease Code	Project Short Name	Days to Last Follow Up
Age at Diagnosis	Site Primary	Gender
Vital Status	Histology	Days to Last Known Alive
Ethnicity	Histological SubType	Sample Type
Person Neoplasm Cancer Status	Program	Project Short Name
Sample Type		Disease Code
Menopause Status		Race
Histological Type		Days to Birth
BMI (Body Mass Index)		Age at Diagnosis
Tobacco Smoking History		Vital Status
Pathologic Stage		Days to Death
HPV Status		Program
Program		Ethnicity
Gender		
Days to Last Known Alive		
Preservation Method		
Project Short Name		
Race		
Tumor Tissue Site		

Favorites Filter

This filter allows the user to add selected variables from existing variable Favorite list.

Clinical Filter Feature Search

This filter allows the user to search by any clinical feature in a given program that is present in the most recent data uploaded for that program.

User Uploaded Programs Filter

This filter allows you to select by filters that you have uploaded using the upload data functionality. It's seperated by projects within your program, a drop down list will display the features that are assocaited.

Selected Filters Panel

This is where the filters you have selected are shown on the right panel for clear verification of what has been selected for analysis. Clicking "Clear All" will remove all selected filters.

Editing a Variable Favorites List

Selecting the edit button from the variable list page or a specific variable details page, you are re-directed to a page where you can add or remove filters from all programs on system and user uploaded data. You are also able to change the title of the variable favortie list from the feature.

Main Menu

- Edit Button: Selecting this menu item make the filters panel appear. And filters selected will be additive to any filters that have already been selected. To return to the previous view, you much either save any selected filters, or choose to cancel adding any new filters.

- Delete Button: Selecting this button will delete you variable favorites list.

- Apply New Workbook button: Selecting this button will create a new workbook with the variable favorites list for analysis.

Selected Filters Panel

This panel displays any filters that have been used on the variable list or any of its ancestors. These cannot be modified and any additional filters applied to the cohorts will be deleted.

Deleting a Variable Favorites List

From the dashboard: Click the arrow next to the variable favorites list a box will appear with the delete option. Confirm the deletion to permanently delete the list.

From within the variable favorites list: If you are viewing the variable favorites list you created, then you delete the cohort by clicking the delete button under the selected variables list.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.7 Working With Cohorts

Cohorts are an way of creating custom groupings of the samples and/or cases that you are interested in analyzing further. You may frequently re-use a cohort in multiple analyses. Creating a "saved cohort" allows you to do this. If you have any existing saved cohorts, they will appear here for you to view, edit and share (see below for details).

Creating and saving a cohort

To create a cohort from Your Dashboard, if you do not have a cohort created, click on the "Filter" or the "Barcodes" links in the "Saved Cohorts" panel at the bottom of the page. The Filter link will take you to the cohort creation page with all filters explained below. The Barcodes link will send you to a page where you can uplaod samples/cases barcodes and create a cohort from them. Currently you can only upload from three programs TCGA, CCLE, and TARGET.

If you already have saved cohorts, they will be listed in the "Saved Cohorts" panel. Click on the "Saved Cohorts" link in that panel and this will take you to a page that displays the details of your saved cohorts. Alternatively, to go directly to a given cohort, click on its name and you will be taken to the cohort details page of that cohort.

To create a new saved cohort, use the "Create Cohort: Filters / Barcodes" link.

Cohort Creation Filters Page

Using the provided list of filters on the left hand side, you can select the attributes and features that you are interested in either from ISB-CGC data or the User Data tab. TCGA Data is the first program to be displayed, next to it is CCLE and TARGET Data tabs. You are able to create a cohort with multiple program filters. CCLE (The Cancer Cell Line Encyclopedia) data - is open access data set that can be used to view sequence data with the IGV viewer without having dbGaP permissions.

By clicking on a feature, the field will expand and provide you with additional filtering options. For example, when you click on "Vital Status", it expands and provides a list containing "Alive", "Dead", and "NA" as options you may choose from. Selecting one or more of these will cause the filter(s) to appear in the Selected Filters panel and visualizations on the page will be updated to reflect that the current cohort has been filtered according to Vital Status. The numbers beside the selectable filter values reflect the number of samples that have that attribute based on all other filters that have been selected.

Individual selections in a filter are "ORed" together, meaning if any of the selected conditions are met they will be in the filter. Filters are "ANDed" together, meaning that selecting two filters means that the cases and samples are created based on both filters being executed. There may be cases where you have 0 cases and samples, because the combination of filters you have chosen are ALL not present (AND function).

Program Selection Panel

The panel in the center of the screen, with four tabs called "TCGA DATA", "CCLE DATA", "TARGET DATA", and "USER DATA" will allow to create a cohort between data programs in the system and data that you have uploaded. The TCGA, CCLE, and TARGET DATA tab each have three tabs called "CASE", "DATA TYPE", and "MOLECULAR" which allow you to apply filters to the cohorts your are creating using ISB-CGC hosted data. For the USER DATA tab, there is one tab called "PROJECTS & STUDIES" which allow you to filter by the projects or studies you have uploaded to the system. Below are the details of each tab.

Please Note: Selecting the program filter will add all samples pertaining to program. Also there is a mouse over feature that will display the disease code long name if it's part of the TCGA, CCLE, or TARGET dataset.

TCGA Cases Tab	TCGA Data Type Tab	CCLE Cases Tab	TARGET Cases Tab	TARGET Data Type Tab
Program	Pathology Image	Program	Program	mRNA Gene Quantification
Project Short Name	Somatic Mutation	Project Short Name	Project Short Name	miRNA Isoform Quantification
Disease Code	Copy Number Segment Masked	Disease Code	Disease Code	miRNA Gene Quantification
Vital Status	mRNA Gene Quantification	Gender	Vital Status	Aligned Reads
Gender	DNA Variation VCF	Sample Type	Gender	
Age at Diagnosis	Aligned Reads	Site Primary	Age at Diagnosis	
Sample Type	Protein Quantification	Histology	Sample Type	
Tumor Tissue Site	miRNA Isoform Quantification	Histological SubType	Race	
Histological Type	miRNA Gene Quantification		Ethnicity	
Pathologic Stage	mRNA Isoform Quantification		WBC at Diagnosis	
Person Neoplasm Cancer Status	Genotypes		Year of Diagnosis	
Neoplasm Histologic Grade	DNA Methylation Beta		Event Free Survival	
BMI (Body Mass Index)			Days to Last Followup	
HPV Status			Days to Last Known Alive	
Residual Tumor			Days to Birth	
Tobacco Smoking History			Days to Death	
Race				
Ethnicity				
Year of Diagnosis				
Menopause Status				
Days to Last Known Alive				
Preservation Method				

Molecular Tab

Gene Mutation Status (creating a cohort based on the presence of a mutation (of various types) in a gene or genes).

This is currently only available for TCGA data. To combine multiple gene filters, select AND (requires all filters to be met for the data to be filtered) or OR (at least one criteria needs to be met for the data to be displayed).

> NOTE 1: If you use AND and do not see the data you are expecting in the filter, try OR instead. AND is a more restrictive criteria requiring all filters to be met, OR is less restrictive, requiring only one criteria

to be met for the data to appear.

NOTE 2: Please add in your saved cohort title the term AND or OR for the combination you used, as the type of combination used in your cohort does not currently appear in the summary of filters in a saved cohort. We are working on this.

You can also filter by different genomic builds and hence different BigQuery Tables. The two BigQuery tables you can select from are *TCGA_hg19_data_v0:Somatic_Mutation_MC3* and *TCGA_hg38_data_v0:Somatic_Mutation*.

Programs & Projects Tab

- User Program
- User Project

Save As New Cohort Button

Push this button if you wish to save the cohort based on the filters you have set. You will be asked for a cohort name and the selected filters will be displayed. Enter the name (any text) and push the "Create Cohort" button.

NOTE: When working with multiple programs you will see a yellow notification box stating, "Your cohort contains samples from multiple programs. Please note that filters will only apply to samples from the program indicated by the tab they were chosen on - they will not apply to samples from other programs in this cohort."

Selected Filters Panel

This is where selected filters are shown for each program so there section to see what filters have been selected. You have to toggle between program tabs to see the filters selected for each program.

If you have not saved the cohort yet, clicking on "Clear All" will remove all selected filters for that program. Also, if you have not saved the cohort yet, selecting an X beside a single filter will remove that filter. If you have saved the cohort, the X is not present as this function is disabled in saved cohorts (to add back to an existing cohort, you can use set operations - see below).

Details Panel

This panel shows the Total Number of Samples and Total Number of Cases in a cohort that is actively being created with the filters that have been selected. If there is a small "timer" icon, the calculation is taking place - the results should appear soon.

Clinical Features Panel

This panel shows a list of images (called "treemaps") that give a high level breakdown of the selected samples for a handful of features for the selected program:

TCGA Clinical Features Panel	CCLE Clinical Features Panel	TARGET Clinical Features Panel
Disease Code	Disease Code	Disease Code
Vital Status	Gender	Vital Status
Sample Type	Site Primary	Gender
Tumor Tissue Site	Histology	Sample Type
Gender	Histological SubType	Age At Diagnosis
Age At Initial Pathologic Diagnosis		

By using the "Show More" button, you can see the last two tree maps. Mousing over an image shows the details of each specific section of the image and the number of samples associated with it.

Data File Availability Panel

NOTE: this function is now available only when editing a created cohort. To access this function, please create the cohort and edit it as described below.

This panel shows a parallel sets graph of available data files for the selected samples in the cohort. The large headers over the vertical bars are data types. Each data type (vertical bar) is subdivided according to the different platforms that were used to generate this type of data (with "NA" indicating samples for which this data type is not available). Each sample in the current cohort is represented by a single line that "flows" horizontally from left to right, crossing each vertical bar in the appropriate segment.

Hovering on a swatch between two vertical bars, you will see the number of samples that have data from those two platforms.

You can also reorder the vertical categories by dragging the headers left and right and reorder the platforms by dragging the platform names up and down.

Programs & Projects Panel

This panel displays a list of images (called "treemaps") similar to the clinical features panel, but can only be found when the User Data tab is selected. This panel displays a high level breakdown of the projects and studies you have uploaded to the system. Another similarity to the clinical features panel hovering over the image will show details of the specific section of the image and the number of samples associated with it.

Cohort Creation Barcodes Page

This feature will allow you upload or enter your own list of sample or cases barcodes from multiple programs. There is a blue instructions button present on both the upload and Enter tabs.

Below is an example of what you are provided when the *Instructions* tab is selected.

All entries must contain a valid case barcode or sample barcode, and a valid program short name.

Valid program short names:

- TCGA
- CCLE
- TARGET

When only a case barcode is provided, all samples from that case will be included in the cohort. Make a separate entry per sample if only specific samples from a case should be added to the cohort.

Please do not include any column headers. Values may be placed in single or double quotes. Format your entries as follows:

Case Barcode,Sample Barcode,Program Short Name

When omitting either barcode type, please be sure to still include the empty column's delimiter. (See examples below.) You may place an empty quoted value ("" or ''), but this is optional.

```
**Example Barcode Set**

TCGA-N9-A4Q4,,TCGA
,TCGA-N7-A4Y8-01A,TCGA
'Saos-2','CCLE-Saos-2', 'CCLE'
'Hs 863.T', '','CCLE'
"TARGET-51-PAJLIV",, "TARGET"
TARGET-51-PAJMFS,"", "TARGET"
```

Upload Tab

This feature allows to upload files with barcodes to create a cohort. Files must be in GDC Data Portal case manifest format, or in comma/tab-delimited case/sample/program format. The file can be a maximum of 32MB. Also, files must be in tab- or comma-delimited format (TSV or CSV) and have an extension of .txt, .csv, or .tsv. After selecting the file and uploading it, the entries will be validated. Any entries which are found to be invalid will be listed, and you can choose to omit them and continue with cohort creation, or select a new file for verification and upload.

GDC Data Portal Case Manifest Files

GDC Data Portal case manifests can be obtained on the 'Cases' tab of the Exploration section of the data portal 'Cases' tab of the Exploration section of the data portal. JSON case manifests must have a .json extension, and will be validated against the GDC's JSON schema. The minimum required properties for each entry in the JSON file are the project object and the submitter_id field. The project object must include the project_id property. All other properties will be ignored.

TSV case manifests must have a .tsv extension, and must contain the first 3 columns of the GDC TSV case manifest in the following order: Case UUID, Case ID, Project. Any other columns will be ignored. Do not remove the header row of the TSV case manifest.

Because the GDC Data Portal case manifest entries are cases, all samples from a case will be included in the cohort.

Enter Tab

This feature will allow you to manually input barcodes for cohort creation. There is a maximum length of 10000 character for the text box. Please use the file upload option if you need to upload more barcodes than will fit in that space.

Operations on Cohorts

Viewing and Editing a Cohort

Once you have created a "Saved Cohort" you can view and edit it. To view a cohort, select it by clicking on its name either from the "Saved Cohorts" panel on the main "Your Dashboard" page or on the "Cohorts" page listing all your saved cohorts.

When you have gone to the "Cohorts" page, you will be shown details of the cohort on the "SAVED COHORTS" tab. The "PUBLIC COHORTS" tab shows public cohorts that are commonly selected. Public cohorts can be used for a "New Workbook" and "Set Operations".

From the "COHORTS" page you can select:

- New Workbook: Pushing this button creates a New Workbook using the selected Cohorts

- Delete: Allows you to delete selected cohort(s) (if you confirm by clicking the second delete button presented)

- Set Operations: Allows you to perform set operations on selected cohorts (see below for details)

- Share: This will share the web view of the cohorts with users you select by entering the users e-mail. If the email address you entered in not registered in the database you are prompted with a message saying, "The following user emails could not be found; please ask them to log into the site first:(email entered)."

Set Operations

You can create cohorts using set operations on the Cohorts page.

To activate the set operations button, you must have at least one cohort selected in your "Cohorts" page. Upon clicking the "Set Operations" button, a dialogue box will appear. Now you may do one of the following:

- Enter in a name for the new cohort you're about to create.

- Select a set operation.

- Edit cohorts to be used in the operation.

- Add A Cohort

The intersect and union operations can take any number of cohorts and in any order. The complement operation requires that there be a base cohort, from which the other cohorts will be subtracted from.

Note: To combine the User uploaded data and the ISB-CGC data, use the Set Operations function. This is possible since the list of barcodes is what is used to create the set operation. For example, to make a cohort of user data samples and ISB-CGC curated samples, Set Union must be used, and to filter user data which is an extension of TCGA or TARGET samples, Set Intersection must be used.

The figure below shows what the results of the set operations will be (represented by I for Intersect, U for Union, and C for Complement). There are two types of sets shown, those that overlap (on the left) and those that are nested (on the right). For the last row (complement operations), the "Subtracted" area is removed from the "Base" area to result in the Complement (C).
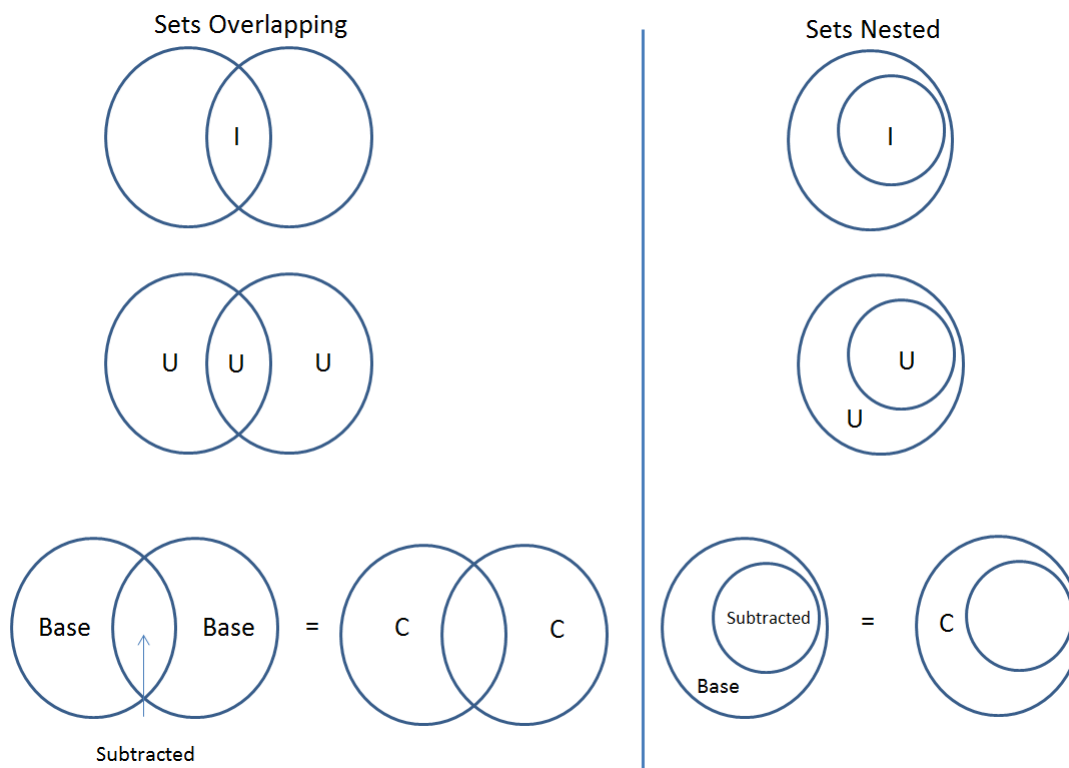
Click "Okay" to complete the set operation and create the new cohort.

Cohort Details Page

The cohort details page displays the details of a specific cohort. The title of the cohort is displayed at the top of the page.

From the "SAVED COHORTS" tab you can:

- New Workbook: Pushing this button creates a New Workbook using the cohort

- Edit: Pushing this button makes the filters panel appear. And filters selected will be additive to any filters that have already been selected. To return to the previous view, you must either save any NEW selected filters (with the "Save Changes" button), or choose to cancel adding any new filters (by clicking the "cancel" link).

- Comments: Pushing "Comments" will cause the Comments panel to appear. Here anyone who can see this cohort can comment on it. Comments are shared with anyone who can view this cohort. They are ordered by newest on the bottom. Selecting the "X" on the Comments panel will close the panel. Any user who owns or has had a cohort shared with them can comment on it.

- Duplicate: Making a copy will create a copy of this cohort with the same list of samples and cases and make you the owner of the copy. This is how you create a copy of another researchers cohort that they have shared with you (note: If they later change their cohort, your cohort will not be updated, it will remain the same as it was at the time you duplicated it).

- Delete: Allows you to delete this cohort (if you confirm by clicking the second delete button presented)

- File Browser: Allows you to view the list of files associated with this cohort (see details below)

- CSV: Provides a list of sample and cases IDs in the cohort

- BigQuery: This will allow you to create a new table or append to an existing table. You must have registered a BigQuery dataset with a Google Cloud Project on the registered Google Cloud Projects details page. More

information on how to register a BigQuery dataset can be found here. If a user wants to export their cohort to a premade table of their own, we require it to have the necessary columns. Here's the schema:

```
{
    'fields': [
        {
            'name': 'cohort_id',
            'type': 'INTEGER',
            'mode': 'REQUIRED'
        },{
            'name': 'case_barcode',
            'type': 'STRING',
            'mode': 'REQUIRED'
        },{
            'name': 'sample_barcode',
            'type': 'STRING',
            'mode': 'REQUIRED'
        },{
            'name': 'project_short_name',
            'type': 'STRING',
            'mode': 'REQUIRED'
        },{
            'name': 'date_added',
            'type': 'TIMESTAMP',
            'mode': 'REQUIRED'
        },{
            'name': 'case_gdc_uuid',
            'type': 'STRING'
        }
    ]
}
```

Note: You shouldn't ever set UUID to 'required' because sometimes a sample doesn't have a UUID, and the attempt to insert a 'null' will cause the cohort export to fail.

- GCS: This will save the details of the cohort in a Google Cloud Storage location specified by you. You must have a registered Google Cloud Storage (GCS) bucket with a Google Cloud Project on the registered Google Cloud Projects details page. More information on how to register a GCS bucket can be found here. You will be able to select the file format to export into Cloud Storage, either CSV or JSON. All files are exported are converted into zip files.

- Share: This will share the web view of the cohorts with users you select by entering the users e-mail. If the email address you entered in is not registered in the database you are prompted with a message saying, "The following user emails could not be found; please ask them to log into the site first:(email entered)."

Current Filters Panel

This panel displays current filters that have been used on the cohort or any of its ancestors. If you have selected multiple These cannot be modified. To add additional filters to this list use the Edit button.

Details Panel

This panel displays the Internal ISB-CGC Cohort ID (the identifier you use to programmatically use this cohort through our APIs), and the number of samples and cases in this cohort. The number of samples may be larger than the number of cases because some cases may have provided multiple samples. This panel also displays "Your Permissions" which

---

can be either owner or reader, as well as revision history. If you have edited the cohort, the filters that were used to originally create the cohort are displayed under the "Creation Filters" label, the newly applied filters since original creation are displayed under the "Applied Filters" label.

TCGA DATA, CCLE DATA, TARGET DATA and USER DATA Tabs

Selecting any program tab will be enabled if you have selected filters for that program. By selecting the tab you will display the Clinical Features panel and the Data File Availability panels for the program selected.

Clinical Features Panel

This panel shows a list of tree maps that give a high level break of the samples for a handful of features for the program view selected:

TCGA Clinical Features Panel	CCLE Clinical Features Panel	TARGET Clinical Features Panel	USER DATA Programs & Projects Panel
Disease Code	Disease Code	Disease Code	Program
Vital Status	Gender	Vital Status	Project
Sample Type	Site Primary	Gender	
Tumor Tissue Site	Histology	Sample Type	
Gender	Histological Sub-Type	Age At Diagnosis	
Age At Initial Pathologic Diagnosis			

Data File Availability Panel

This panel shows a parallel sets graph of available data files for the selected samples in the cohort. The large headers over the vertical bars are data types. Each vertical bar may be broken up to represent different platforms used to generate that type of data (and "NA" for samples for which that data type is not available). The sets of lines that "flow" from left to right indicate the number of samples for which each type of data files are available. If you hover over a horizontal segment between two bars, you will see the number of samples that have both those data type platforms. You can also reorder the vertical categories by dragging the headers left and right and reorder the platforms by dragging the platform names up and down.

File Browser Page

"File Browser" takes you to a new page where you can view the complete list of data files associated with your current the cohort. The file list page provides a paginated list of files available with all samples in the cohort. Here, "available" refers to files that have been uploaded to the ISB-CGC Google Cloud Project, including both controlled access and open access data. You can use "Show", "Page", "Previous" and "Next" navigation tools to navigate to more values in the list. The columns are sortable by selecting the column header. You can select a subset of the default columns to show by using the "Choose Columns to Display" tool.

You can filter by Genomic Build either HG19 or HG38 and view which platforms and files are available for the build selected. You can enter a case barocde on all tabs as a filter option. You also have the ability to do a partial string search. To remove the search key word you click the clear "X" button. After useing the case search button the filter count is updated for the rest of the filters. You may also filter on these files if you are only interested in a specific data

type, data format, platform, disease code, disease strategy, and/or experimental strategy. Selecting a filter will update the associated list. The numbers next to the filter refers to the number of files available for that filter.

The tabs "IGV", "Pathology Images" and "Radiology Images" allow you to filter for files that show you respectively read-level sequence data (viewed using the IGV viewer), pathology images, and radiology images. Please note: only if you have authenticated as a dbGaP authorized user will you be able to select controlled access files to view in the IGV viewer (CCLE data does not require authorization to view the sequence data in the IGV viewer). Details of how to view Sequences, and Pathology and Radiology images are provided below.

Download File List as CSV

To download a list of files that are part of this cohort, select the button in the upper right on the File Listing panel (on all tabs) called "CSV". This will begin a download process of all the files available for the cohort, taking into account the selected filters.

The file contains the following information for each file:

- Case Barcode
- Sample Barcode
- Program
- Platform
- Experimental Strategy
- Data Category
- Data Type
- Data Format
- Genomic Data Commons(GDC) File UUID
- Google Cloud Storage(GCS) location
- Genomic Data Commons(GDC) Index
- Index File Google Cloud Storage(GCS) location
- File Size
- Access Type (open or controlled access)

Export File List to BigQuery

To export the File list to BigQuery, select the button BigQuery. You will need to have registered a Google Cloud Project and a BigQuery dataset to be able to export to BigQuery. More information on how to register a BigQuery Dataset can be found here. You can either make a new table or append an existing table. You can also give the table a unique name if left blank we will provide a name for the table.

The table will contain the following information (for each of the data type tabs):

- row
- cohort_id
- case_barcode
- sample_barcode
- project_short_name

- date_added

- build

- gdc_file_uuid

- gdc_case_uuid

- platform

- exp_strategy

- data_category

- data_type

- data_format

- cloud_storage_location

- file_size_bytes

- index_file_gdc_uuid

- index_file_cloud_storage_location

Export File List to Google Cloud Storage

To export the File list to Google Cloud Storage (GCS), select the button GCS. You will need to have registered a Google Cloud Project and a GCS Object to be able to export to GCS. More information on how to register a GCS bucket can be found here. You can also give the object a unique name if left blank we will provide a name for the bucket. You can either choose to export as JSON or CSV file and all files exported are converted into zip files.

The file will contain the following information (for each of the data type tabs):

- sample_barcode

- case_barcode

- cloud_storage_location

- file_size_bytes

- platform

- data_type

- data_category

- exp_strategy

- data_format

- gdc_file_uuid

- gdc_case_uuid

- project_short_name

- cohort_id

- build

- index_file_storage_location

- index_file_gdc_uuid

- date_added

Viewing a Sequence

When available, sequences in a cohort can be viewed using the IGV viewer. To find those sequences that can be viewed with the IGV viewer, open a cohort and select the "View Files" button at the top of the page. The files associated with your cohort will be shown, a tab displaying the files that can be viewed with the IGV viewer. Selecting the check-box in the "View" column (for a maximum of 5 files) and the "Launch IGV" button in the upper window will take you to an IGV view of the selected sequence(s) data. Controlled access files will be viewable by sequence ONLY if you have authenticated as a dbGaP-authorized user.

(more information about Viewing a Sequence in the IGV Viewer).

Using the Image Pathology Viewer

When available, pathology images can be viewed using the caMicroscope tool (see more about caMicroscope provide here). These are the pathology images that are associated with TCGA samples (not all files are currently available, due to some metadata that is not in place at GDC describing the image files. ISB-CGC is working with GDC to resolve this issue, and more images will be appearing when that issue is resolved). To find images that can be viewed, open a saved cohort and select the "File Browser" button at the top of the page. You can also select the File Browser button from the Dashboard Saved Cohorts panel. The files associated with your cohort will be shown, with the third tab listing any available Pathology Images. Hovering over the File Name column and clicking on 'Open in caMicroscope' will open the image file in a new tab using caMicroscope (HINT: using a smaller cohort will provide faster response in creating the list of files available).

To zoom into the image, either click the left button or use your wheel to zoom in. Use your mouse to move around the image. To zoom out of the image, shift-slick the left mouse button or use your wheel to zoom out. Selecting caMicroscope at the top of page will send you to the caMicroscope homepage. If you hover over the Slide Barcode section on the top right hand side you will see metadata information listed.

Viewing a Radiology Image

To find images that can be viewed, open a saved cohort and select the "File Browser" button at the top of the page. You can also select the File Browser button from the Dashboard Saved Cohorts panel. The files associated with your cohort will be shown, with the fourth tab listing any available Radiology Images. Hovering over the Study Instance UID column and clicking on 'Open in Osimis Web Viewer' will open the series Selection panel in a new tab using Osimis DICOM (HINT: using a smaller cohort will provide faster response in creating the list of files available).

For a more detailed step-by-step process of Viewing Radiology Images using the Osimis DICOM viewer please go here.

Deleting a cohort

From the "COHORTS" page: Select the cohorts that you wish to delete using the checkboxes next to the cohorts. When one or more are selected, the delete button will be active and you can then proceed to deleting them.

From within a cohort: If you are viewing a cohort you created, then you can delete the cohort using the delete button on the menu.

Creating a Cohort from a Visualization

To create a cohort from visualization, you must be in plot selection mode. If you are in plot selection mode, the crosshairs icon in the top right corner of the plot panel should be blue. If it is not, click on it and it should turn blue.

Once in plot selection mode, you can click and drag your cursor of the plot area to select the desired samples. For a cubbyhole plot, you will have to select each cubby that you are interested in.

When your selection has been made, a small window should appear that contains a button labelled "Save as Cohort". Click on this when you are ready to create a new cohort.

Put in a name for you newly selected cohort and click the "Save" button.

Copying a cohort

Copying a cohort can only be done from the cohort details page of the cohort you want to copy.

When you are looking at the cohort you wish to copy, select Duplicate from the top menu.

This will take you to a new copy of the cohort.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.8 Program Data Upload

Uploading your own data is a way of creating custom groupings of the samples and/or cases that you are interested in analyzing further with the data that is already preexisting in our system or tools that we have on the system. You may frequently re-use the data that was uploaded in multiple analyses. Creating a "Program" allows you to do this. If you have any existing Programs with data uploaded, they will appear here for you to view, edit and share (see below for details).

Files and File Formats

The Program Data Upload uses a number of pre-defined file formats to get data into the system and make it available for use. The **Other/Generic** file format is the most flexible. This format assumes that the first row of the file contains the column headers and all subsequent rows contain data. The remaining file formats are all matrix formats where the first column (or columns in some data types) contain identifiers like gene or miRNA name, the first row contains sample identifiers and the "cells" contain the actual data values. Examples of the accepted matrix format files are shown below:

NOTE: For the matrix files, the text case matters for the required columns (lower case is different from upper case). In addition, the ISB-CGC system will not validate any identifiers such as barcodes or gene names. It is up to the user to make sure that uploaded data is correctly identified.

- DNA Methylation

  This is a simple matrix file. The first column should have the header **Probe_ID**. Sample barcodes should be the headers for all remaining columns.

Probe_ID	Barcode 1	Barcode 2	Barcode N
Probe ID 1	Value 1	Value 2	Value N
Probe ID 2	Value 1	Value 2	Value N
Probe ID N	Value 1	Value 2	Value N

- Gene Expression

  The Gene Expression matrix file has two required columns:

– **Name**: This is the accession number for the gene

– **Description**: This is the gene symbol for the gene

Name	Description	Barcode 1	Barcode 2	Barcode N
Accession 1	Gene name 1	Value 1	Value 2	Value N
Accession 2	Gene name 3	Value 1	Value 2	Value N
Accession N	Gene name N	Value 1	Value 2	Value N

- microRNA

   There is one required and one optional column for microRNA:

   – **miRNA_ID** is required and is generally the ID for the miRNA_ID

   – **miRNA_name** is optional and can be used to provide alternative names for the miRNA. If not present, the BigQuery data table will have **null** in this column

miRNA_ID	miRNA_name	Barcode 1	Barcode 2	Barcode N
miRNA ID 1	Alt name 1	Value 1	Value 2	Value N
miRNA ID 2	Alt name 2	Value 1	Value 2	Value N
miRNA ID N	Alt name N	Value 1	Value 2	Value N

- Protein Expression

   Protein Expression has three required columns:

   – **Protein_Name**: This is the name or symbol for the protein

   – **Gene_Name**: This is the name of the gene associated with the protein

   – **Gene_Id**: This is the accession number for the gene

Protein_name	Gene_Name	Gene_Id	Barcode 1	Barcode 2	Barcode N
Protein 1	Gene Name 1	Gene ID 1	Value 1	Value 2	Value N
Protein 2	Gene Name 2	Gene ID 2	Value 1	Value 2	Value N
Protein 3	Gene Name 3	Gene ID 3	Value 1	Value 2	Value N

- Other/Generic

Files in Other/Generic format are not matrix files, but rather have the data in columns. The order of the columns is very flexible, and the upload interface will allow users to define what kind of data is in each of the columns. The only requirement is that one, and only one, of the columns should be sample barcodes. In addition, all rows must have the same number of columns. Any completely blank columns will be flagged and should be removed. Any columns containing blank entries will have *null* used for the blanks in the BigQuery data table.

NOTE: Currently, each Sample Barcode can only be represented once in a file. Files with the same barcode on multiple rows will cause a failure. If you have multiple data values for a single barcode (like gene expression values for multiple genes) you will either have to create a matrix file or upload multiple files to Other/Generic.

Creating and Saving a New Program

To create a new program from Your Dashboard, if you do not have a program created, click on the "Upload Program Data" link in the "Saved Programs" panel at the bottom of the page. This will take you to the Data Upload page.

If you already have Programs created, they will be listed in the "Saved Programs" panel. Click on the "Saved Programs" link in that panel and this will take you to a page that displays the details of your existing Programss. Alternatively, to go directly to a given Program, click on its name and you will be taken to the program details page of that program.

Registering Cloud Storage Buckets and BigQuery Datasets - a pre-requisite for using your own data in ISB-CGC

You will need to have a BigQuery Dataset and a Google Cloud Storage bucket registered to your Google Cloud Project through the Google Project details page in the UI. (Please note: the names of the buckets and datasets are case sensitive.)

How To Register Buckets and Datasets

Once you have created a bucket and a dataset in the Google Cloud Console of your Google Cloud Project, you will need to register them with your project name using the Webapp.

Step 1: Click on your user icon in the upper right.
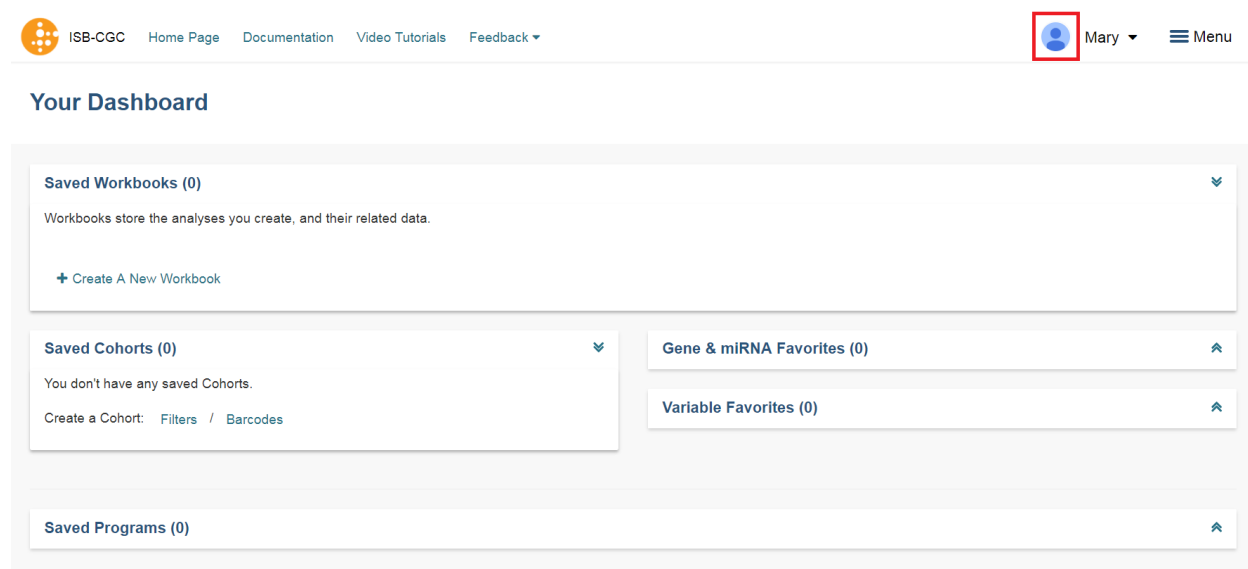
Step 2: Click on "View Registered Google Cloud Projects"

Step 3: Click on the project you wish to use. If you have not registered a project, follow the instructions **'here'_**.

Step 4: Use the "Register Cloud Storage Bucket" or "Register BigQuery Dataset" links to add buckets and datasets as needed

Data Upload Page

A New Program

To start an entirely new program, users should click on the **Upload Program Data** link on the front page of the Webapp (*Your Dashboard*). This will bring up a form where a new program can be defined. Users should fill out the required fields and any optional fields that would be helpful. Clicking on **Select File(S)** button will bring up a dialog to select the file with data.

NOTE: You can upload multiple files in a single step. The **Type** drop-down should be used to indicate what data type the file represents. If the data type is one of the choices besides **Other**, the file will have to conform to the specifications listed at the top of this *page*. For a more complete description of the options on this page, see the *Data Upload Page Components* section.

Project description and file selection

Clicking on the **Next** button brings up a form where users will select which bucket and BigQuery dataset the file upload should use. These buckets and datasets were *registered* according to the process above. The **Platform** and **Pipeline** fields can contain any useful description a user wishes to provide.

A New Project For An Existing Program

Adding a new project to an existing program follows the same steps as creating a new program. However, instead of filling out the new program information fields, users should click on the **A New Project For An Existing Program** tab and select an existing program from the drop-down menu. All other steps for describing and uploading the file will remain the same.

Data Upload Page Components

This section describes the features found on the Data Upload page.

Sharing User Uploaded Programs

This will share the web view of your uploaded program with users you select by entering the users e-mail. The User will receive an e-mail message with a link to your shared uploaded program explaining that (you) wanted to share a program with (them) and that you have invite them to join. If the email address you entered in not registered in the

database you are prompted with a message saying, "The following user emails could not be found; please ask them to log into the site first:(email entered)."

System Data Dictionary Link

This link goes to the System Data Dictionary which is a comprehensive list of all clinical data fields and possible values. This dictionary can be helpful in aligning metadata from the imported program to ISB-CGC data fields.

High Level Data Files

High level data files usually represent some level of data analysis as opposed to raw files. High level files can be used in Workbooks and visualized alongside ISB-CGC data.

Low Level Files for API Access

Files uploaded as low-level files for API access will not be usable in the Webapp, but rather will appear in the user's Google Storage Bucket. This feature is intended for files like BAM or VCF files that contain more raw data.

File Type

This is the data type that the uploaded file represents. Currently the allowed data types are:

- Gene Expression
- miRNA Expression
- Protein Expression
- Methylation
- Other

File Format Requirements

All files must be tab delimited and meet the formatting requirements described in *Files and File Formats*.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.9 Graphing User Data

Once a user has uploaded their own data to the Webapp, that data can be vizualized using the same graphing tools that are available when graphing TCGA and/or TARGET data. However, the process for graphing user data is slightly different from how it is done with TCGA data

Important sections on the Webapp front page

The boxes in the figure below are links that are used to graph user data

Step 1: Create a Cohort from your project

- From the front page of the Webapp, click on Create Cohort to start the process
- Click on the *User Data* tab and select the project or study that will be the cohort
- Save as a new cohort

Step 2: Create a Variables Favorite

- From the front page of the Webapp, click on *Create Variable Favorites* to start the process

- Click on the *Projects* tab to see the user supplied studies

- Select the variables that will be available to graph. Note that if the study has a large number of selections, using the browser search function can help locate the item.

- Give the variables a name and click on the *Save as Favorite* button

Step 3: Graph the favorites in a Workbook

- From the front page of the Webapp, click on *Create a new Workbook*

- Under the *Source Data* heading, select the Variables and Cohorts that you wish to use in the graph. In each case you will be brought to a page listing all of the available Variables or Cohorts. Simply select the desired ones and then click the *Add to Workbook* button

- Under the *Analysis Type* heading, select the appropriate graph type. This will cause a window to slide in from the right.

- Fill in the X and Y axis variables, select a variable to use for coloring and finally select the cohort to use.

- Click on the *Update Plot* button to have the system gather the data and generate the plot.
- If changes need to be made to the plot, click on the *Edit Analysis Settings* link to bring back the graph dialog box.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.10 Integrative Genomics Viewer (IGV)

IGV is a widely used interactive tool for exploring genomic data. A web-based version is integrated into the ISB-CGC Web-App, and the IGV desktop version can also be used to access ISB-CGC hosted data in Google Cloud Storage (GCS). Information about this use-case is provided in this section. For more information about IGV, please follow the links in the Acknowledgements section at the bottom of this page.

Accessing the IGV Browser from the Web Application

To access IGV, first select a cohort and then go to the cohort file list page (through the "View Files" link at the top of the page).

Your Dashboard > Saved Cohorts >

CCLE program

| New Workbook | Edit | Comments | Duplicate | Delete | File Browser | CSV | BigQuery | GCS | Share | Shared With (0) |

Current Filters

CCLE

Program: CCLE

Cohort Details

Internal ISB-CGC Cohort ID: 1793

Total Number of Cases: 948

Total Number of Samples: 953

Your Permissions: OWNER

Show More
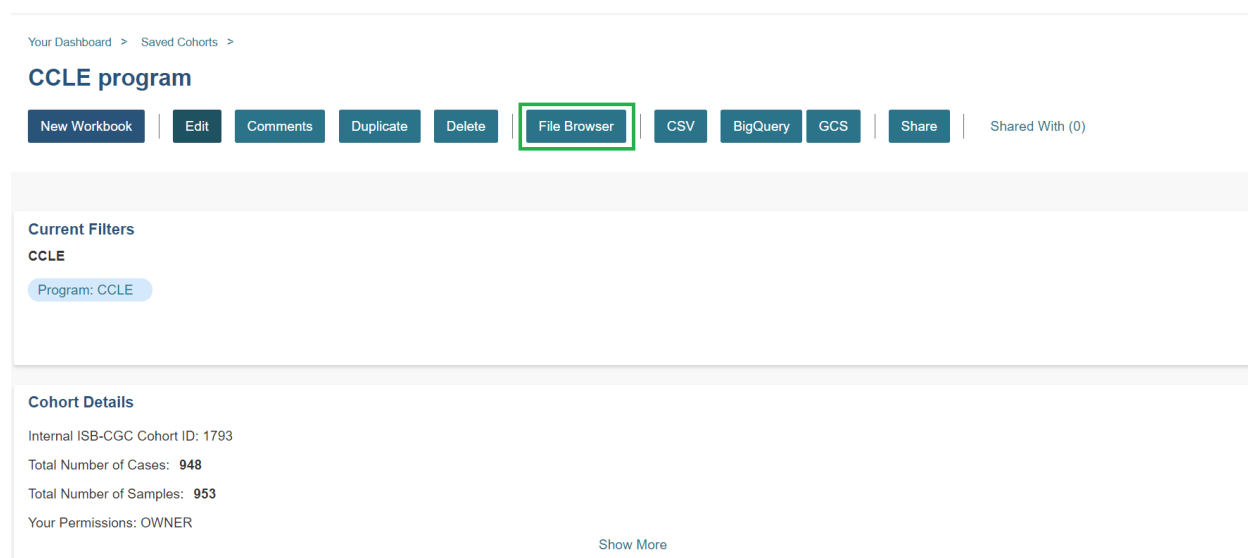
The resulting file list can be filtered using the build either HG19 or HG38 and the Platforms listed on the left. Any file that can be displayed in the IGV Browser will have a "Cloud Storage" (for files available via Google Cloud Storage) checkbox in the IGV column on the right side of the file table. Note that many files viewable in IGV may require that the user have dbGaP authorization to view controlled access data. If the user has been authenticated and authorized through the user details page, the user will be able to select files. Otherwise the cursor will be disabled when the user hovers over a checkbox. Open source data such as the CCLE project do not require dbGaP authorization and can be viewed by any authenticated user.

Once a maximum of five files have been selected, they can be viewed in the IGV Browser by clicking on the "Launch IGV" button in the upper right of the window

Your Dashboard > Saved Cohorts > CCLE program >

File Browser: CCLE program

| All Files | IGV | Pathology Images | Radiology Images |

Build

HG19 ▾

▸ CASE

▸ DATA TYPE

▸ DATA FORMAT

▸ PLATFORM

▸ DISEASE CODE

▸ DATA CATEGORY

▸ EXPERIMENTAL STRATEGY

Selected BAM Files   [Launch IGV]

Note: Files viewed in IGV must all be for the same build.

2 file(s) selected (limit 5)

CCLE-BFTC-905, RNA-Seq, Illumina HiSeq, Aligned reads [HG19] ×   CCLE-SW 780, RNA-Seq, Illumina HiSeq, Aligned reads [HG19] ×

File Listing   [CSV] [BigQuery] [GCS]

Showing 1 to 25 of 1273 entries

Show 25 ▾ entries   Page [   ] [Go]   Previous **1** 2 3 ... 51 Next

Choose Columns to Display ▾

Program ▲	Case Barcode	File Name	Disease Code	Exp. Strategy	Platform	Data Category	Data Type	Data Format	View
CCLE	BFTC-905	G27311.BFTC-905.1.bam [GDC ID: 5d0a2d39-5d2...	BLCA	RNA-Seq	Illumina HiSeq	Raw sequencing	Aligned reads	BAM	☑

NOTES:

- You will only be able to view controlled access sequence files if you have logged in as a registered dbGaP authorized user.

- You will need to disable your browser pop-up blocker to view files with IGV. If you see a 403 error when using the IGV viewer, the pop-up blocker is the cause of that error. Turn off the blocker and try again.

Using IGV Desktop Application to View Aligned Reads in Google Cloud Storage

You can also download and use the IGV desktop application to view aligned reads stored BAM files in Google Cloud Storage. To do this, download the most recent version of IGV. After launching IGV, go to the "Settings" menu to enable the Google Menu item in the application (directions on how to do this).

To load BAM files from ISB-CGC Google Cloud Storage, use the "File" > "Load from URL..." menu item in the IGV application, entering the path to the bam file in GCS. Paths to BAM files stored by ISB-CGC can be found using the `cohorts().cloud_storage_file_paths()` and `samples().cloud_storage_file_paths()` APIs described here.

NOTE:

- You will only be able to view controlled access sequence files if you have logged in as a registered dbGaP authorized user.

Acknowledgements

The copyright to the Integrative Genomics Viewer is held by the Broad Institute, and the software has been released under the MIT License. For more information about IGV please see the IGV home page or the IGV github repo.

We are grateful to the IGV team for their assistance in integrating IGV into the ISB-CGC web-app.

Robinson J T, Thorvaldsdottir H, Winckler W, Guttman M, Lander E S, Getz G & Mesirov J P, *Integrative genomics viewer*, Nature Biotechnology 29, 24-26 (2011).

Thorvaldsdottir H, Robinson J T, Mesirov J P, *Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration*, Briefings in Bioinformatics 14, 178-192 (2013).

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.11 Viewing and using cohorts in the Webapp and API

Cohorts are one of the central concepts that researchers use when analyzing large datasets. As has been discussed elsewhere in the documentation, cohorts can be created either in the Webapp or via the ISB-CGC REST API. What may not be as clear is that cohorts created by one of the systems can be viewed and used in the other. In other words, you can create a cohort using the API and use it in the webapp or you can create a cohort in the webapp and use it in the API. This can give researchers significant flexibility in creating and sharing their cohorts.

It should be noted that the details of how to use the APIs can differ significantly depending on how users access the REST APIs. The examples given here are assuming users only have access to a console and not a higher-level language like Python where the APIs can be used more programatically. Additionally, the examples shown here are using the TCGA endpoint, but exactly the same functionality is aviailable for TARGET and CCLE using the endpoints specific to those programs.

Related documents:

- Creating Saved Cohorts in the Web Application
- Details of "Cohorts. . . APIs" in the ISB-CGC API documentation

Listing Cohorts

Any cohort you've previously created can be seen using either the Webapp or the API. In the Webapp, cohorts can be viewed both on the front page of the app as well as on the Cohorts page as shown in Figure 1. For ease of use, there is A) a shortcut to go to a cohort's "File browser" page (A in the top diagram), and B) a "Collapse/Expand" arrow (B in the top diagram) to collapse or expand a long list of items in your lists as needed.

Figure 1: Cohorts shown on the front page (top) and Cohorts page (bottom) in the Webapp

Similarly, the **cohorts().list()** endpoint of the ISB-CGC API will return exactly the same set of cohort. The Google API Explorer is a convenient tool for examining API output from within a browser. In addition to showing the data returned from a query, it will also show a constructed query that can be used in a script (see Figure 2).

Figure 2: Google API Explorer

Creating Cohorts

Creating cohorts using the Webapp has been fully documented and needs no further explanation. Creating cohorts using the API uses two different endpoints, cohorts().preview() and cohorts().create(). These two endpoints have exactly the same query capabilities and differ only in that the preview endpoint will return the results of the query without creating a cohort while the create endpoint will create the cohort and name it using the name provided with the name attribute. In addition, due to the authentication requirement for the create endpoint, the query is sent as a JSON object

In the following example, the first query creates a cohort of patients from the UCS and CESC studies who were 20 years old or younger at the time of diagnosis. Since this query is run against the **preview** endpoint, no cohort is actually created, only the results shown in Figure 3 are returned.

```
https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/cohorts/preview?age_
→at_initial_pathologic_diagnosis_lte=20&project_short_name=TCGA-UCS&project_short_
→name=TCGA-CESC
```

```
https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/cohorts/preview?age_
↪at_initial_pathologic_diagnosis_lte=20&project_short_name=TCGA-UCS&project_short_
↪name=TCGA-CESC
```

Figure 3: Using Google API Explorer to preview creating a cohort

Due to the need for authentication and cohort naming, actually creating the cohort requires some modifications of the preview query. First, the *name* attribute needs to be specified with the name users will see in both the Webapp and in the **cohorts().list()** endpoint.:

```
https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/cohorts/create?name=
↪{COHORT NAME}
```

Additionally a JSON object containing the query needs to be created.

```
{"Study": ["TCGA-UCS", "TCGA-CESC"], "age_at_initial_pathologic_diagnosis_lte": 20}
```

The commands above will create a cohort via the API

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.12 Radiology Viewer

Radiology images are viewed in an Osimis Web Viewer, a plug-in to the Orthanc Image Server (Orthanc). The ISB_CGC web application uses an instance of Orthanc to manage radiology files for the purpose of viewing. Currently only DICOM formatted files from TCGA samples are available for viewing. It may be helpful to review the DICOM Model of the Real World to understand the relationship between patients DICOM studies, DICOM series and DICOM instances.

The ISB-CGC web application File Browser page presents a table of DICOM studies associated with patients in some cohort.

Viewer Components

Clicking on a study in the table opens an Osimis Web Viewer in a new tab:

All the DICOM series which comprise the selected study are shown as thumbnail images in the Series Selection Panel. Note that it can take several seconds for these thumnails to appear. By default, the thumbnails are laid out in a grid pattern. This can be changed to a list format by clicking on the **List Display** button above the thumbnails. The list format displays a description of each series.
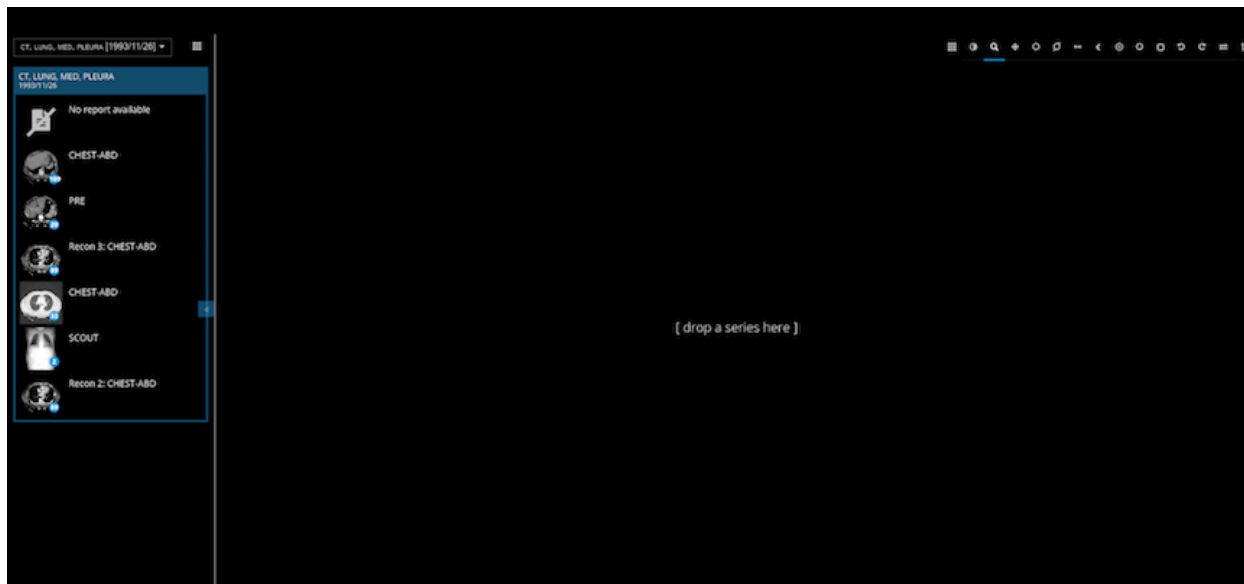
Change back again to the grid pattern by clicking on the **Grid Display** button .

To the lower right of each thumbnail is a small blue circle in which is displayed the number of DICOM instances which comprise the corresponding DICOM series. In addition, when your cursor hovers over a thumbnail, the viewer cycles through the instances comprising the series. (Note these low resolution images are loaded in the background and may not be available for cycling immediately after the viewer window opens.)

To view a larger rendering of a series, drag its thumbnail into the viewport. The first instance of the series is immedi-

ately displayed.
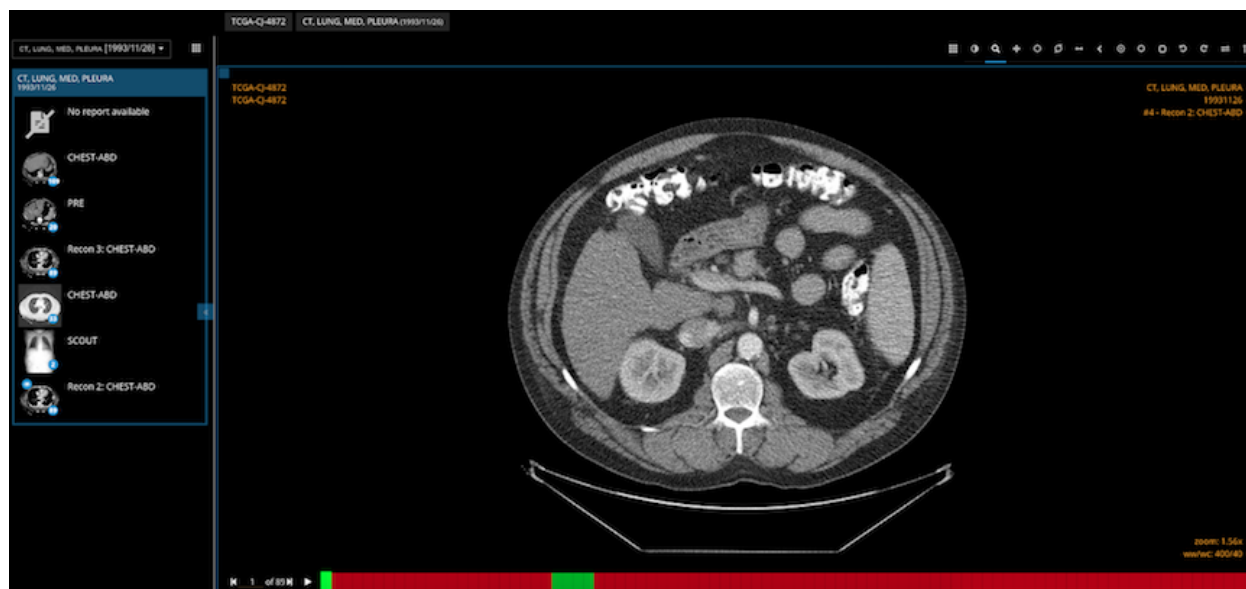
At the same time, at the bottom of the viewport you will notice a grid comprised of a series of rectangles corresponding to the instances in the series. The color of the tabs indicates the following:

- Black: The corresponsing instance is not yet avialable for viewing

- Red: A reduced resolution image of the instance is available for viewing

- Green: The full resolution image of the instance is available for viewing

Typically, the viewer loads reduced resolution images for all series as quickly as possible. It loads full resolution images only when a series is dragged into the viewport.

When instance images have been loaded, you can scroll through the instances using your mouse's thumbwheel or equvalent. As you scroll, the grid at the bottom of the screen highlights the instance currently being displayed. Clicking on a rectangle in the grid causes the corresponding instance to be displayed. The **Play Controls**
 in the lower left corner of the main window enable you to single step forward or back through the series, and to cycle throught series repeatedly. A frame rate slider pops up when you hover over the play button.

Viewing Functions

A set of buttons above the viewport provides a range of functions.

The **Layout** button controls subdividing the viewport for the simultaneous display of one, two or four series. Drag a series into any of the subviewports to display it. Clicking in a subviewport gives it focus for mousewheel and cursor drag operations.
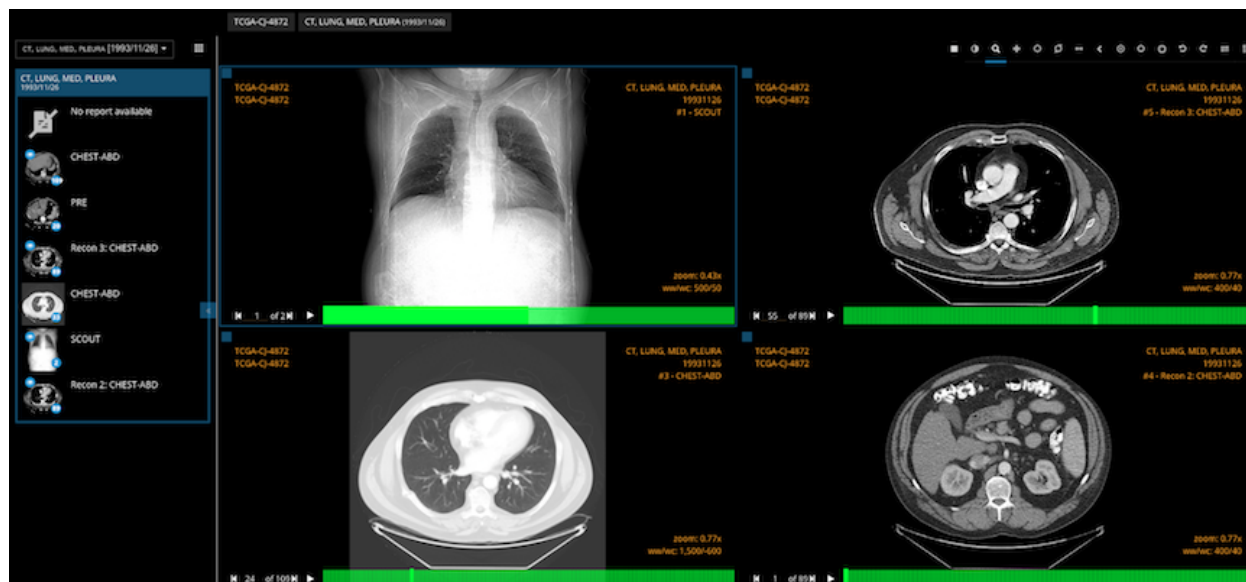
Of the remaining buttons, some are modal, changing the effect of the cursor drag function. A blue line underscores the currently selected mode. Other buttons immediately perform some operation on the subviewport that has focus.

- The **Invert Color** button immediately inverts the colors of the series in the (sub)viewport having focus.

- The **Zoom** button is modal. When selected, dragging the cursor with mouse button depressed expands or contracts the series in the (sub)viewport having focus. Expansion/contraction is around the cursor position at which dragging begins.

- The **Pan** button is modal. When selected, dragging the cursor with mouse button depressed causes panning of the series in the (sub)viewport having focus.

- The **Windowing Presets** button operates both modally and immediately. Hovering the cursor over the button displays a list of windowing presets, one of which can be selected by clicking on it. The selection immediately sets Window Width (WW) and Window Center (WC) values for the series in the (sub)viewport having focus. The WW,WC value pair specifies a linear conversion from stored pixel values to values to be displayed. See **here_** for further information on Window Center and Window Width.

  DICOM instances generally include WW,WC value pairs and these are used by default. Other WW,WC value pairs that may be appropriate for specific cases can be selected on the pop-up. The *Preset #1* selection restores WW,WC to the DICOM setting.

  The Windowing Presets button operates modally when clicked. In this mode, dragging the cursor left or right in a (sub)viewport changes the Window Width value applied to the series in that (sub)viewport. Dragging the cursor up or down in a (sub)viewport changes the Window Center value applied to the series in that (sub)viewport.

- The **Magnifying Glass** button is modal. Hovering the cursor over the button displays a pop-up containing two sliders that control the magnification level and size of a virtual magnifying glass. When selected, dragging the cursor with mouse button depressed opens a virtual magnifying glass that displays a magnified rendering of the underlying image in the region of the cursor.

- The **Length Measurement** button is modal. When selected, the distance in physical units between two points in an instance can be measured. To perform a measurement, click the mouse button once with the cursor over some point of interest, and then again over a second point of interest. Alternatively, depress and hold the mouse button while the cursor is over the first point of interest, then release the mouse button while the cursor is

over the second point of interest. A line joining the two points and its length are displayed. The line will scale if the series is zoomed in or out.

A length measurement can be moved by clicking on it and dragging. To remove a length measurement, drag it or an endpoint outside of the extent of the between instance. Note that if you have "zoomed in" on an instance, its extent may be much larger than the (sub)viewport in which it is displayed. This can make it difficult to drag the measure outside of the extent of the instance. In this case it may be necessary to "zoom out" in order to be able to drag the measure outside of the extent of the instance.

A length measurement is only visible on the instance on which it was made. There is currently no support for saving length measurements.

- The **Angle Measurement** button is modal. When selected, the angle between features in an instance can be measured. To perform a measurement, click on a point of interest in an instance. A pair of lines are displayed. Drag the end points of the lines as needed to form the angle to be measured. The angle between the lines is displayed continuously as any endpoint is dragged.

  An angle measurement can be moved by clicking on one of the lines and dragging it while holding down the mouse button. To remove an angle measurement, drag it or an endpoint outside of the extent of the instance. Note that if you have "zoomed in" on an instance, its extent may be much larger than the (sub)viewport in which it is displayed. This can make it difficult to drag the measure outside of the extent of the instance. In this case it may be necessary to "zoom out" in order to be able to drag the measure outside of the extent of the instance.

  An angle measurement is only visible on the instance on which it was made. There is currently no support for saving angle measurements.

- The **Pixel Probe** button is modal. When selected, clicking on a point in an instance displays a circle at the probe point, the X and Y location of the pixel relative to the top left corner of the instance, and the intensity or color of the selected pixel. The value of color instance pixels is specified in RGB coordinates. For monochrome instances, both a Stored Pixel value (SP) and a Modality Pixel value (MO) are displayed. The MO values is calculated as `SP * RescaleSlope + RescaleIntercept`, where RescaleSlope and RescaleIntercept are DICOM values of the instance.

  A pixel probe can be moved by clicking on the probe indicator and dragging it while holding down the mouse button. To remove a pixel probe, drag it outside of the extent of the instance. Note that if you have "zoomed in" on an instance, its extent may be much larger than the (sub)viewport in which it is displayed. This can make it difficult to drag the measure outside of the extent of the instance. In this case it may be necessary to "zoom out" in order to be able to drag the measure outside of the extent of the instance.

  A pixel probe is only visible on the instance on which it was made. There is currently no support for saving pixel probes.

- The **Elliptical ROI** button is modal. When selected, click on an instance and drag either of the small circles to configure an elliptical region of interest. The area, in pixels, of the ellipse is displayed near the ellipse. On monotone instances, the mean and standard deviation of the intensities of the pixels within the ellipse are also displayed.

  An ellipse can be moved by clicking on its border and dragging it while holding down the mouse button. To remove an elliptical ROI, drag the ellipse or one of its control points outside of the extent of the instance. Note that if you have "zoomed in" on an instance, its extent may be much larger than the (sub)viewport in which it is displayed. This can make it difficult to drag the ROI outside of the extent of the instance. In this case it may be necessary to "zoom out" in order to be able to drag the ROI outside of the extent of the instance.

  An elliptical ROI is only visible on the instance on which it was made. There is currently no support for saving elliptical ROIs.

- The **Rectangle ROI** button is modal. When selected, click on an instance and drag either of the small

circles to configure a rectangular region of interest. The area, in pixels, of the rectangle is displayed near the rectangle. On monotone instances, the mean and standard deviation of the intensities of the pixels within the rectangle are also displayed.

A rectangle can be moved by clicking on its border and dragging it while holding down the mouse button. To remove a rectangular ROI, drag the rectange or one of its control points outside of the extent of the instance. Note that if you have "zoomed in" on an instance, its extent may be much larger than the (sub)viewport in which it is displayed. This can make it difficult to drag the ROI outside of the extent of the instance. In this case it may be necessary to "zoom out" in order to be able to drag the ROI outside of the extent of the instance.

A rectangular ROI is only visible on the instance on which it was made. There is currently no support for saving rectangular ROIs.

- The **Rotate Left** button immediately performs a ninety degree left rotation of the image in the (sub)viewport that has focus.

- The **Rotate Right** button immediately performs a ninety degree right rotation of the image in the (sub)viewport that has focus.

- The **Flip Horizontally** button immediately performs a flip about the Y axis of the image in the (sub)viewport that has focus.

- The **Flip Vertically** button immediately performs a flip about the X axis of the image in the (sub)viewport that has focus.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.5.13 Web-App Release Notes

- **April 25, 2019** v3.18

  Please Note: Work is underway to rework our cohort creation page to better display images associated with samples.

  The following datasets (open and controlled access) have been added to the ISB-CGC for service account registration:

  1. The National Cancer Institute Center for Cancer Research (NCICCR)

  2. Foundation Medicine(FM)

  3. Clinical Trial Sequencing Project (CTSP)

  4. Veterans Research for Precision Oncology Program (VAREPOP)

Issues resolved in sprint 33 as of 4/25/2019

New Enhancements

- When working with Oncogrid, OncoPrint, or a SeqPeek plot on a workbook you will receive an automatic list of genes ready for analysis.

- When on a workbook additional text has been added to guide the user to select edit plot settings to select a gene/miRNA/variable filter and cohort to used in analysis selected.

- The Workbook comments section has been reformatted to better align with analysis displayed.

- On the cohort creation - filter page the filters have been updated in the left filter panel to specify the count type displayed (samples).

Bug Fixes

- Clicking on a legend entry to toggle display of the data points on a scatter or violin plot will now work correctly, even if the legend text has a space.

- Plotting with sample type filter on a workbook will now display counts correctly.

- When working with the color by feature on either a Scatter plot or a Violin plot, the numerical values are now displayed as a color-gradient legend.

- When using a workbook with OncoGrid analysis you are now able to plot using genomic build hg19.

- When using a workbook with a Cubby Hole plot analysis text is no longer cut off when using sample type or residual tumor as a filter.

Known Issues in Sprint 33 as of 4/25/2019

- Analysis Type: Seq peek Formatting is Elongated on occasion

- If the user shares a Cohort, neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale, it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When uploading TARGET files using the cohort barcode creation feature from the GDC, you may get an invalid barcodes error message and unable to upload all the barcodes.

- On the File Browser page for Diagnostic images there is no GDC file UUID associated to them.

- Sharing a workbook with someone else will cause the analysis to reset.

- **March 8, 2019** v3.17

Please Note: Work is underway to rework our cohort creation page to better display images associated with samples.

We will soon add the following datasets(open and controlled access) to ISB-CGC:

1. The National Cancer Institute Center for Cancer Research (NCICCR)

2. Foundation Medicine(FM)

3. Clinical Trial Sequencing Project (CTSP)

4. Veterans Research for Precision Oncology Program (VAREPOP)

Is there a dataset that is a priority for you? Of this list which should we include first? If yes, please send all requests to feedback@isb-cgc.org

Issues resolved in Sprint 32 as of 3/07/2019

New Enhancements

- When working with a workbook many overall enhancements of user functionality have been improved.

- Cubby hole plot analysis has been reformatted to better suit the end user by now allowing resizing and scrolling through the cubby hole plot analysis.

- You are now able to work on a workbook via fullscreen for added comfort.

- You are also now able to download plot data for Bar charts, Histogram charts, Scatter plots, Violin plot charts, and Cubby hole plots as a CSV file.

- OncoGrid has been added as an analysis option when working with a workbook.

- On the File Browser section you are now able to use full screen on all image viewers.

- On the register/adjust a service account page, we've clarified the notification message if a key or role is found associated to a service account.

Bug Fixes

- When using a workbook you will no longer see text overlap when working on a violin/scatter plot with the color by feature sample type as filter option.

- When working on the Pathology images viewer you will no longer see text overlap on the top right hand side of viewer.

Known Issues in Sprint 32 as of 03/07/2019

- Analysis Type: Seq peek Formatting is Elongated on occasion

- If the user shares a Cohort, neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale, it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When uploading TARGET files using the cohort barcode creation feature from the GDC, you may get an invalid barcodes error message and unable to upload all the barcodes.

- On the File Browser page for Diagnostic images there is no GDC file UUID associated to them.

- Sharing a workbook with someone else will cause the analysis to reset.

- **January 22, 2018** v3.16

Please Note: Work is underway to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

Issues resolved in sprint 31 as of 01/22/2019

New Enhancements

- On the Gene list creation page you can now upload line separated and tab separated gene lists to be used for analysis.

- We have made some updates to the workbooks plotting section. You are now able to redraw to the original plot after any changes. Also, you are now able to save the plot as a .SVG, .PNG, or a .JSON file.

Bug Fixes

- On the cohort creation using the barcode upload feature, the table page list feature now is now displayed properly.

- If you have not linked to the Data Commons Framework at all you are able to unregister a Google Cloud Project. If you are not linked to the Data Commons Framework, but others in the Google Cloud project are, only they will be able to unregister the GCP.

Known Issues in Sprint 31 as of 01/22/2019

- Analysis Type: Seq peek Formatting is Elongated on occasion

- If the user shares a Cohort, neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale, it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When uploading TARGET files using the cohort barcode creation feature from the GDC, you may get an invalid barcodes error message and unable to upload all the barcodes.

- On the File Browser page for Diagnostic images there is no GDC file UUID associated to them.

- Sharing a workbook with someone else will cause the analysis to reset.

- **December 5, 2018** v3.15

Please Note: Work is underway to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

Issues resolved in sprint 30 as of 12/05/2018

New Enhancements

- The ISB-CGC homepage has been updated to provide Funding and Partnership information, and the About Us section is now hidden by default.

- An introduction video has been added to the videos tutorials section. This video covers the user interface, BigQuery and using the API endpoints.

- Funding information has been updated on the ISB-CGC homepage.

- On the Register/Adjust a service account page all spacing issues have been addressed.

- On the Register/Adjust a service account pages you are now returned more detailed information. You will be returned verification results for all users on the Google Cloud Project, datasets permissions verification, registered service account verification results, and all service accounts verification results.

- On the File Browser page, when working with on a cohort with CCLE data included for genomic build hg38 you are displayed a notification message for CSV export button.

- On the File Browser a new column has been added for File Size for all tabs.

- When exporting a large cohort on the File Browser page you are returned a notification message stating cohort export is underway to check BigQuery in a few minutes.

- On the File Browser you are now able to view/download/print Pathology Reports in pdf format.

- On the Pathology Images viewer, the GDC has released multiple versions of slide barcodes. To handle this we now sort the pathology image files by UUID.

- On the the File Browser for Radiology Images, ISB-CGC has upgraded the viewer to run OHIF for better performance times and views.

Bug Fixes

- When working on the File Browser export to BigQuery/Google Cloud Storage entering an invalid name will disable the export feature, even after toggling between datasets.

- When on a Workbook, using an OncoPrint analysis using certain genes with no gene positions will return correct error message stating no internal feature ID was found.

- Certain gene names which symbol '_' included will now return data points when working with a Workbook.

Known Issues in Sprint 30 as of 12/05/2018

- Analysis Type: Seq peek Formatting is Elongated on occasion

- If the user shares a Cohort, neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale, it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When uploading TARGET files using the cohort barcode creation feature from the GDC, you may get an invalid barcodes error message and unable to upload all the barcodes.

- On the File Browser page for Diagnostic images there is no GDC file UUID associated to them.

- Sharing a workbook with someone else will cause the analysis to reset.

- **September 20, 2018** v3.14

Please Note: Work is underway to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

Issues resolved in sprint 29 as of 09/20/2018

New Enhancements

- When on the File browser page, the case barcode column is included when downloading the file manifest CSV format option.

- You will now need to log into the Data Commons Framework to be able to access controlled data.

Bug Fixes

- API endpoint cohort.creation will no longer include NULL values in sample counts when cohort is created.

- On the File Browser tab using filter option NA will now return all entries associated to it.

- Program TCGA and TARGET have new miRNA based on the GDC release 11 is now available in Google BigQuery and for plotting.

Known Issues in Sprint 28 as of 07/31/2018

- Analysis Type: Seq peek Formatting is Elongated on occasion

- If the user shares a Cohort, neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale, it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When uploading TARGET files using the cohort barcode creation feature from the GDC, you may get an invalid barcodes error message and unable to upload all the barcodes.

- – On the File Browser page for Diagnostic images there is no GDC file UUID associated to them.

- – Sharing a workbook with someone else will cause the analysis to reset.

- – When using a workbook, a gene with symbol "_" will produce a error message saying, "There was an error retrieving plot data. Please try again."

- **July 31, 2018** v3.13

Please Note: Work is underway to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

Issues resolved in sprint 28 as of 07/31/2018

New Enhancements

- – When working on the File Browser you now have the ability to search by case barcode all on tabs(Pathology Images, Radiology Images, IGV Browser, All Files).

- – On the File Browser page for the Pathology Images tab, you can now also filter by Disease Code, Data Format, and Data Type. For the Radiology Images, a disease code was added.

- – On the File Browser page, you now have the ability to hide the filters and expand the file list to full width.

- – On the File Browser page, if you download the file manifest using the export CSV feature, you will see newly updated file paths. The older paths are still in existence but will be deleted within the next month.

- – On the File Browser page if you use a cohort with CCLE data present, switch to build hg38 and attempt to export you will return a notification no CCLE data will be present for build hg38.

- – On the homepage, we have added a carousel scrolling feature for all how-to videos for easy access.

- – A description has been added to all video tutorials.

- – The menu bar text variable favorites have been updated to be undifferentiated.

Bug Fixes

- – When creating a cohort using the filter selection option, if the filter options selected add up to zero the save cohort button will be disabled.

- – A workbook with user upload data and public data e.g TCGA data will plot any analyses.

- – For the export to GCS and BigQuery feature the export button will now disable when an invalid name is given.

- – On a registered Google Cloud Project detail page, datasets can no longer be duplicated within a project, and bucket names are globally unique (across all projects).

Known Issues in Sprint 28 as of 07/31/2018

- – Analysis Type: Seq peek Formatting is Elongated on occasion

- – If the user shares a Cohort, neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- – CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- – When a user duplicates a Worksheet, then tries to implement the log scale, it will not function properly.

- – The set operation for existing Cohorts complement is behaving exceptionally slow.

- – The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- – When uploading TARGET files using the cohort barcode creation feature from the GDC, you may get an invalid barcodes error message and unable to upload all the barcodes.

- API endpoint cohort.creation will include NULL values in sample counts when the cohort is created.

- On the File Browser page for Diagnostic images, there is no GDC file UUID associated to them.

- Sharing a workbook with someone else will cause the analysis to reset.

- When downloading the CSV file for Radiology Images tab on the File Browser page you will noticed there are no samples barcodes associated to Radiology Images. ISB-CGC will add a case barocde to the CSV file export table in the next release.

- **June 18, 2018** v3.12

Please Note: Work is underway to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

Issues resolved in sprint 27 as of 06/18/2018

New Enhancements

- The ISB-CGC has enabled OncoPrint visualization tool for germline mutations (codebase obtained with permission from cBioPortal) as another Workbook analysis tool. For more information please go here.

- You are now able to view Radiology Images from TCIA data through the File Browser using the Osimis viewer. For more information please go here here.

- Two new videos have been added to our video tutorials section. You can now learn how to sign up with a Google account and how to make a gene list easily. For more information please go here. here

- The Dashboard has been upgraded to include a collapse feature for all panels (workbooks and cohorts are opened by default) and a direct link to the File Browser has been added to the Cohorts panel.

- Under cohort creation by filters, the Molecular tab for TCGA data has been upgraded to combine multiple gene mutation filters. Filters can be combined using AND (requires all filters to be met for the data to be filtered) or OR (at least one criteria needs to be met for the data to be displayed).

- The CSV download, Export to BigQuery, and Export to GCS feature has been added to the IGV Browser, Pathology Images, and the Radiology Images tab on the File Browser.

- On the File Browser All files tab the clinical filter now displays the accurate count available for analysis.

- The File Browser has been upgraded to now include the option of which columns to display and the ability to jump to any page.

- The site menu has been improved to allow faster load times and better overall performance. Please Note that Workbooks must now be created from a data source (Cohorts, Variable lists, Gene & miRNA lists) or from the Workbook list page.

Bug Fixes

- When working on Firefox browser a violin plot will display the data plotted correctly when working on a Worksheet.

- A cohort with user uploaded data present and public data present in our system e.g TCGA data, the cohort details page for the selected filters panel will sort the filters by their appropriate program.

- On the cohort creation - barcode upload page the 'Samples' and 'Cases' column headers were sometimes swapped. This has been corrected.

- When trying to reload a stored Seq-Peek plot from a Workbook the previous gene selection is stored and the plot will automatically be loaded.

- On the File Browser IGV Browser tab when switching genomic builds the view column selection option will be disabled.

Known Issues in Sprint 27 as of 06/18/2018

- Analysis Type: Seq peek Formatting is Elongated on occasion

- If the user shares a Cohort, neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale, it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When uploading TARGET files using the cohort barcode creation feature from the GDC, you may get an invalid barcodes error message and unable to upload all the barcodes.

- API endpoint cohort.creation will include NULL values in sample counts when cohort is created.

- Duplicate entries can be entered for the register a dataset and the register a bucket on the Google cloud project details page.

- On the File Browser page for Diagnostic images there is no GDC file UUID associated to them.

- Sharing a workbook with someone else will cause the analysis to reset.

- A Workbook using a cohort that has user uploaded data and public TCGA data present will not return data for any analysis.

- **May 3, 2018** v3.11

Please Note: Work is underway to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

Issues resolved in sprint 26 as of 05/03/2018

New Enhancements

- The export to BigQuery feature has been enhanced to include faster processing time for larger cohorts with e.g 30,000 > samples and 65,000 > file records.

- You are now able to export cohort and cohort file manifests to a Google Cloud Storage using either .JSON or .CSV format from the cohort details page and from the File Browser page.

- We have enhanced our instructions associated with buttons to further provide directions to the end-users.

- On the File Browser page it is now possible to change how many entries are displayed at a time, as well as sort columns by clicking on the column header.

- Google Cloud Project membership is now automatically updated every six hours. If you are adding someone new to the project they will be able to use the project after six hours maximum without someone having to log in and manually refresh the project.

Bug Fixes

- You can no longer share a cohort with yourself (email currently logged into) and cause the file browser page to disable.

- DNA methylation has been re-enabled to be used with hg38 and hg19 data when working with workbooks and plotting.

- Sharing inputs have had their security restrictions tightened. This also includes the registering a service account page.

- On the File Browser page when downloading the file manifest via the CSV button you are no longer able to re-select the CSV button while the file is building.

- On the File Browser tab if you toggle between entries pages on the All Files tab it will not affect the IGV tab or Pathology Images tab entries counts display.

- On the File Browser page you can now freely toggle between entries pages with no errors displayed.

- On the File Browser page selecting filters from the left hand side while exploring pages will no longer crash and require you to back or refresh the page to fix.

Known Issues in Sprint 26 as of 05/03/2018

- Analysis Type: Seq peek Formatting is Elongated on occasion

- If the user shares a Cohort, neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale, it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When working on Firefox browser a violin plot does not display the data plotted correctly when working on a Worksheet.

- When uploading TARGET files using the cohort barcode creation feature from the GDC, you may get an invalid barcodes error message and unable to upload all the barcodes.

- API endpoint cohort.creation will include NULL values in sample counts when cohort is created.

- Duplicate entries can be entered for the register a dataset and the register a bucket on the Google cloud project details page.

- A cohort with user uploaded data present and public data present in our system e.g TCGA data, the cohort details page for the selected filters panel does not properly display the filters selected.

- On the File Browser page for Diagnostic images there is no GDC file UUID associated to them.

- **April 2, 2018** v3.10

Please Note: Work is underway to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

Issues resolved in sprint 25 as of 04/02/2018

New Enhancements

- When working with the File List table you can now Export the cohort file list to BigQuery for later analysis.

- When registering or adjusting a service account to use controlled data, the page will no longer briefly appear as if no datasets had been selected. This should reduce confusion.

- Selecting the refresh project button from a registered Google Cloud Project details page will leave you on the details page rather than redirecting you to the registered Google cloud project list table page.

- On the cohort creation page, using the barcode upload page, the valid/invalid entries table can now be sorted by on any column with either ascending/descending order.

- Removing someone from the IAM and Admin list does not remove them from the web-app automatically. If the removed user still has the GCP present in their webapp interface attempting to register or refresh a service account will remove the GCP from the web app, and a display message informing them they are no longer a member of the project will be seen.

- When working with any tables that can be sorted on smaller screens, there is no longer any text overlap in the table columns.

- Character restrictions has been relaxed, you can now use characters such as []{}(); for entity names and descriptions.

Bug Fixes

- SeqPeek and CNVR can only be plotted with TCGA data, but if a cohort contains no TCGA samples the SeqPeek analysis will now return an error message saying, "The chosen cohorts do not contain samples from programs with Gene Mutation data."

- API endpoint samples.get can now be used to return data for all three programs.

- On the adjust service account page, when attempting to remove the service account from being able to access controlled data, and then immediately trying to add the service account back to controlled data, the system will require you to verify the service account's users again.

Known Issues in the Sprint 25 as of 04/02/2018

- Analysis Type: Seq peek Formatting is Elongated on occasion

- If the user shares a Cohort, neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale, it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When working on Firefox browser a violin plot does not display the data plotted correctly when working on a Worksheet.

- When uploading TARGET files using the cohort barcode creation feature from the GDC, you may get an invalid barcodes error message and unable to upload all the barcodes.

- On the cohort File List Browser page, while you are downloading CSV files, other filters can be selected.

- **February 28, 2018** v3.9

Please Note: Work is underway to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

Issues resolved in Sprint 24 as of 02/28/2018

New Enhancements

- On the register a Google Cloud Project you now can only register the project ID. Registering the project name or project number will now result in an error message. Additionally, the GCP Project Name and ID will now both display on the GCP detail and list pages, and refreshing a GCP Project in the Web Application will update the Name if it was changed in the GCP console.

- For cohort creation via sets of barcodes, the barcode set (pasted in the text box or uploaded as a file) can now be a simple list of sample or case barcodes separated by newlines, commas, or tabs;

the program listing is no longer needed, and you don't need to supply the barcodes in a distinct columnar format.. The previous 3-column format will continue to work as well.

- On a worksheet, if no table is being searched the BQ table(s) used panel becomes inactive.

Bug Fixes

- When editing the name of a cohort the cancel feature is now working properly.

- When working on a worksheet the SeqPeek feature will now work with all genes.

- All genes can be plotted on a worksheet when working with a histogram.

- When registered Service Accounts for controlled data, the Adjust/Register can only be clicked once.

- When working with SeqPeek, the BQ table(s) used panel will now refresh every time even if no new data is plotted.

- When a user is removed from their Google project the user interface doesn't remove the project from their list. Instead, the individual removed will receive error messages saying they are no longer on the project if they try to refresh the project or register the service account.

- On a registered Google Cloud Project page, the refresh button will now properly add and remove users from the project if they are added or removed from the IAM and Admin list on the Google console.

- When working on the Internet Explorer you can again create a cohort using the filter creation page.

- When using the dbGaP eRA authentication you will now be logged out at 24 hours instead of 16 hours.

- For cohort creation when uploading a large set of barcodes you will no longer return a 400 bad request error.

Known Issues in Sprint 24 as of 02/28/2018

- Analysis Type: Seq peek Formatting is Elongated on occasion

- If the user shares a Cohort, neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale, it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When working on Firefox browser a violin plot does not display the data plotted correctly when working on a Worksheet.

- When uploading TARGET files using the cohort barcode creation feature from the GDC, you may get an invalid barcodes error message and unable to upload all the barcodes.

- SeqPeek and CNVR can only be plotted with TCGA data, but if a cohort contains no TCGA samples the SeqPeek analysis will still search the TCGA BigQuery tables

- API endpoint samples.get currently down and will return a 503 error for all three programs.

- On the File Browser page, while you are downloading CSV files, other filters can be selected.

- **February 1, 2018** v3.8

Please Note: Work is underway to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

Issues resolved in Sprint 23 as of 02/01/2018

New Enhancements

- We have enabled DNA methylation data to be used when plotting with genomic build hg38.

- The cohort view files page has been updated to File Browser. The File Browser page also now has new filters data level, data type, disease code, data format, and experimental strategy. A time stamp has also been added to the CSV file that can be downloaded.

- The IGV browser and caMicroscope are now more clearly defined and separated on the File Browser page.

- When uploading a set of barcodes to create a cohort the error message has been redefined to direct someone to the instructions.

Bug Fixes

- You can now plot DNA methylation data using genomic build hg19 when working on a worksheet.

- When registering a service account to controlled data you will no longer receive an error message when certain Google managed service accounts are also on the IAM and Admin page.

- On a worksheet, if you add new cohorts to a worksheet with pre-existing cohorts. Now the older and newly added cohorts are present on the worksheet for analysis.

- When working with a worksheet you are now able to plot gene names that contain periods.

Known Issues in Sprint 23 as of 02/01/2018

- You cannot make a cohort using the cohort creation filter option on an Internet Explorer browser.

- Analysis Type: Seq peek Formatting Elongated on occasion.

- If the user shares a Cohort neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When working on Firefox browser a violin plot does not display the data plotted correctly when working on a Worksheet.

- When uploading TARGET files using the cohort barcode creation feature from the GDC you may get an invalid barcodes error message and unable to upload all the barcodes.

- SeqPeek can only be plotted with TCGA data, but if a cohort contains no TCGA samples the SeqPeek analysis will still search the TCGA BigQuery tables.

- API endpoint samples.get currently down and will return a 503 error for all three programs.

- Currently unable to use TARGET data with the IGV browser to view .bam files.

- When editing the name of a cohort the cancel feature is not working properly.

- When working on a worksheet the SeqPeek feature is currently not working with certain genes.

- Certain genes will produce a blank chart with no data on a worksheet when working with a histogram.

- **December 20, 2017** v3.7

Please Note: Work is underway to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

Issues resolved in Sprint 22 as of 12/20/2017

New Enhancements

- Using the 'View Files' page you can now view TCGA pathology images using caMicroscope!

- After logging into dbGaP you are now redirected to the user details page.

- Due to recent updates with Google, we have implemented new security requirements when working with the service accounts and attempting the access the controlled data. For more information about new requirements please go here.

Bug Fixes

- You will no longer experience a 502 error when trying to create a new variable favorite list if you have uploaded a lot of your own data using the user data upload feature.

Known Issues in Sprint 22 as of 12/20/2017

- Analysis Type: Seq Peek formatting elongated on occasion

- If the user shares a Cohort neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When working on Firefox browser a violin plot does not display the data plotted correctly when working on a Worksheet.

- When working on a workbook if you add new cohorts to the worksheet the pre-existing cohorts will be de-selected from the worksheet.

- If you have uploaded a lot of data using the User Data Upload feature, it is likely you will experience 502 error page when attempting to create a new variable favorite list.

- When uploading TARGET files using the cohort barcode creation feature from the GDC you may get an invalid barcodes error message and unable to upload all the barcodes.

- **November 20, 2017** v3.6

Issues resolved in Sprint 21 as of 11/20/2017

New Enhancements

- You can now send a cohort you have created in the web application to a new BigQuery dataset or append an existing table.

- The cohort creation by uploading barcodes feature has been extended to include .JSON and .TSV files from the Genomic Data Commons data portal.

- Created a new API endpoint to be used to return a GCS object URL given a GDC file identifier also known as a UUID.

- Updated the registered Google Cloud Project to clearly state if the project's service accounts are active or not.

- You can now enter special characters into the comments section for workbooks and cohorts e.g URL

- On the register a service account page the Compute Engine default service account is automatically added to the enter service ID text box.

- When creating a new cohort we have implemented a text saying, "Creating cohort…" for instances when creating a new cohort takes a little longer than usual.

- We have significantly sped up loading times for the cohorts detail and cohorts table list page for users who have 50 + cohorts which caused slow loading time.

Bug Fixes

- A duplication of the exact cohort will no longer happen when you select the confirmation multiple times while the page is loading working with Set Operations.

- On the cohort details, you can no longer select the clinical feature panel and edit filters without selecting the edit button first.

- On the cohort creation page, you can use the clinical feature panel to select filters when working with the User data upload tab.

Known Issues in Sprint 21 as of 11/20/2017

- Analysis Type: Seq peek Formatting Elongated on occasion

- If the user shares a Cohort neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When working on Firefox browser a violin plot does not display the data plotted correctly when working on a Worksheet.

- When working on a workbook if you add new cohorts to the worksheet the pre-existing cohorts will be de-selected from the worksheet.

- If you have uploaded a lot of data using the User Data Upload feature, it is likely you will experience 502 error page when attempting to create a new variable favorite list.

- When working with the API endpoints the sample.get for all three programs will return a 503 internal server error.

- **October 13, 2017** v3.5

Please Note: We need to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

Issues resolved in Sprint 20 as of 10/11/2017

New Enhancements

- You can now upload sample and case identifiers from programs TCGA, CCLE and TARGET to create a cohort.

- We have begun to allow the addition/removal of a service account with a new button instead of the user having to re-register the service account every time.

- For the Set Operations feature when working with cohorts has been enhanced and has become easier to work with.

- For the Set Operation Complement feature you will now create a cohort faster than before.

- You will now be displayed mouse over text when working with the New Workbook, Delete, Set Operations, and Share button on the Cohorts list details page.

- The About Us link in the top left of the page has been re-named to Homepage.

Bug Fixes

- All bam files for the TARGET program are available to be used with the IGV browser.

- On the Cohort creation page, you can now select a filter for your Cohort by selecting an option from the Clinical Feature graphs using Histological Type for program CCLE.

Known Issues in Sprint 20 as of 10/12/2017

- Analysis Type: Seq peek Formatting Elongated on occasion

- If the user shares a Cohort neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- A duplication of the exact cohort happens when you select the confirmation multiple times while the page is loading working with Set Operations.

- The mouse-over feature is currently disabled for program TARGET with disease code ALL.

- When working on Firefox browser a violin plot does not display the data plotted correctly when working on a Worksheet.

- **September 21,2017** v3.4

  Please Note: We need to rework our cohort creation page to better differentiate between samples which are from image data vs. those which are not.

  Issues that have been resolved in Sprint 19 as of 09/21/2017

  New Enhancements

  - When plotting, certain values will now be displayed as categorical when before it was displayed as a numerical value e.g Tobacco Smoking History.

  - The Homepage has been updated to incorporate links for TARGET and CCLE programs.

  - The extended list of programs and projects on the new User Uploaded Data creation page is now displayed in alphabetical order.

  - On the user details page you are now shown a confirmation box when you attempt to unlink the NIH identity account associated to the Google Identity you originally logged in with.

- When working with Workbooks you are now shown a table on the top right hand side of Worksheet which shows what BigQuery tables the information being displayed is from.

- On the Cohort creation page you can now select a filter for your Cohort by selecting an option from the Clinical Features graphs.

- On the user details page, if you attempt to associate you Google Identity to an NIH Identity that is already registered in the system to another Google Account you are given a yellow error message stating which email the NIH Identity is already associated to.

Bug Fixes

- When working with Workbooks the log scale graphing option will be saved when a user comes back to the Worksheet at another time.

- On the existing Cohorts table list page, the confirmation delete 'blue x' button will now remove a selected Cohort if you select another option e.g Set Operation.

- The Google Cloud Project details page refresh wheel and delete icon are now working properly for service accounts.

- The Cloud Project details page now lists the authorized datasets active with an associated service account.

- When deleting a User Uploaded program you are now sent to the existing programs list page if you delete the program. If you delete the project you stay on the program details page.

- The ownership of a Variable list, Gene and miRNa list, and User Uploaded Programs are now verified. This means you can no longer view any existing in system if you are not the original creator.

- A confirmation on the Register a Service Account page has been implemented for service accounts when the user attempts to register.

- On the Cohort creation when toggling between the tabs for the different programs, you now cannot switch tabs until the tab on display is loaded.

Known issues in sprint 19 as of 09/21/2017

- Analysis Type : Seq peek Formatting Elongated on occasion

- If the user shares a Cohort neither the owner nor the person who was granted access to Cohort will receive a confirmation email when sharing a Cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a Worksheet, then tries to implement the log scale it will not function properly.

- The set operation for existing Cohorts complement is behaving exceptionally slow.

- A duplication of the exact cohort happens when you select the confirmation multiple times while the page is loading working with Set Operations.

- The mouse over feature is currently disabled for program TARGET with disease code ALL.

- A very small amount of bam files for program TARGET currently have the wrong file name and cannot be used with the IGV browser.

- When working on Firefox browser a violin plot does not display the data plotted correctly when working on a Worksheet.

- **August 23, 2017**: v3.3

Issues that have been resolved in sprint 18 as of 08/23/2017

New Enhancements

- Users with NIH-approved access can now view and analyze TARGET (Therapeutically Applicable Research To Generate Effective Treatments) controlled data using service accounts and also on the IGV browser.

- You will be returned a more detailed error message when invalid characters are used with user data uploading titles.

- On the File list page you will be allowed to select only one genomic build at a time for clarity on which build will be used by the IGV browser.

- When attempting to duplicate the registration of your Google Cloud Project you are given an error message saying, "A Google Cloud Project with the id xxx-xxx-xxxx already exists."

- If you attempt to register a service account with the same datasets it already has activated, you will be given an error message saying, "Service account xxxxxxxxxxxx-compute@developer.gserviceaccount.com already exists with these datasets, and so does not need to be registered."

- The Data Use Certification and Agreement covering your access to all controlled data has been added to the user details page in the interface.

- The CCLE user.get API endpoint has been removed from the system due to the fact we do not currently host any controlled CCLE data.

- The format of CSV file downloaded with Download IDs button from the cohort details page has been changed to display the case and sample barcodes as two separate columns.

- From the User uploaded program detail page, you can now edit the project name and description by selecting the gear option.

Bug Fixes

- When creating a large cohort you are no longer returned a red error message.

- The sharing feature for Workbooks, Cohorts, and User Uploaded Programs has been re-activated. You must enter a valid email address that is present in the system to share the workbook, cohort, or user uploaded program. If they are not present in our system please feel free to invite them to the ISB-CGC website.

- When working with a new worksheet or a duplicate worksheet with workbooks for categorical features e.g bar chart, you can no longer select the log option. The log option only applies to numerical options.

- When working with workbooks, selecting the Delete button multiple times will no longer result in an error, and instead return you to the Workbooks list page after successful deletion of the Workbook.

- Users can plot user uploaded data when working with workbooks when using variables and cohorts from the same files that were uploaded.

- The cohort.list API endpoint will display the correct cases count for cohorts listed.

- The Download File List as CSV on the File List page will download the correct information when genomic build hg38 is selected.

- You are no longer able to add XSS-vulnerable characters to the edit section for user uploaded data.

- An improved error message is displayed when attempting to register a Google Project you are not associated with.

- Making a new Gene and miRNA set from a Workbook will no longer result in lowercase gene and miRNA names.

- The TCGA Sample.get API endpoint will no longer return a response with sample ID duplicates.

Known issues in sprint 18 as of 08/23/2017

- Analysis Type : Seq peek Formatting Elongated on occasion

- If the user shares a cohort neither the owner nor the person who was granted access to cohort will receive a confirmation email when sharing a cohort.

- CCLE data cannot be plotted when working with workbooks. ISB-CGC will resolve this functionality after the GDC formally releases CCLE data.

- When a user duplicates a worksheet, then tries to implement the log scale it will not function properly.

- On the existing cohorts table list page, the confirmation delete 'blue x' button does not remove selected cohort if you select another option e.g Set Operation. The same issue can be found in reverse if you select the 'blue x' on the confirmation page for set operation you can then select the delete button and see the cohort on the confirmation panel.

- When working with working with workbooks the log option is not working properly for the plot settings.

- The set operation for existing cohorts complement is behaving exceptionally slow.

- A duplication of the exact cohort happens when you select the confirmation multiple times while the page is loading working with Set Operations.

- When plotting, certain values will be displayed as numerical when it should be a categorical value e.g Tobacco Smoking History.

- The mouse over feature is currently disabled for program TARGET with disease code ALL.

- **July 31, 2017**: v3.2

Please Note: When creating large cohort you will be given a red error message saying, "There was an error saving your cohort; it may not have been saved correctly." This issue is planned to be resolved in the next sprint.

Issues that have been resolved in sprint 17 as of 07/31/2017

New Enhancements

- You will be returned a more detailed error message when using invalid characters when working with user data uploading titles.

- On the File list page you will are allowed to select only one genomic build at a time for better clarification of which build you will view on the IGV browser.

Bug Fixes

- When working with Swap Values button on a worksheet, the log option selected for either axis is now carried over as well when the swap values button is selected.

- On the IGV browser when working with TCGA data build hg38 the interface will no longer return a No feature found with name "efgr" at the bottom of the IGV browser page.

- When working with the cohort.create API endpoint you have the ability to create a large cohort with the barcode filter without a timeout error.

- When creating a cohort with the cohort.create API endpoint you can view the list of barcodes from the cohort details page in the ISB-CGC user interface irrelevant of size.

- When working with the create a new variable favorites list page, you can now create a variable list using the USER DATA tab.

Known issues in sprint 17 as of 07/31/2017

- The sharing feature for Workbooks, Cohorts, and User Uploaded Programs is currently disabled

- Analysis Type : Seq peek Formatting Elongated on occasion

- The CCLE data in GUI is not parallel to the CCLE data in BigQuery.

- Cannot plot any data if you use a CCLE data cohort on a worksheet.

- On the existing cohorts table list page, the confirmation delete 'blue x' button does not remove selected cohort if you select another option e.g Set Operation. The same issue can be found in reverse if you select the 'blue x' on the confirmation page for set operation you can then select the delete button and see the cohort on the confirmation panel.

- The set operation for existing cohorts complement is behaving exceptionally slow.

- A duplication of the exact cohort happens when you select the confirmation multiple times while the page is loading working with Set Operations.

- When working with a new worksheet or a duplicate worksheet with workbooks for categorical features e.g bar chart you can select the log option. The log option only applies to numerical options.

- When working with workbooks, if you select the delete confirmation button multiple times while the page is loading you will be sent to an error page.

- You currently cannot plot user uploaded data when working with workbooks.

- When plotting, certain values will be displayed as numerical when it should be a categorical value e.g Tobacco Smoking History.

- The mouse over feature is currently disabled for program TARGET with disease code ALL.

- The cohort.list API endpoint will display the incorrect cases count for cohort listed.

- The Download File List as CSV on the File List page downloads the wrong information when genomic build hg38 is selected.

- You are currently able to add non-whitelist characters to edit section for user uploaded data.

- You are returned a vague error message on the register a Google Cloud Project page when attempting to register a Google Project you are not associated to.

- The samples and cases filters have not been removed from the cohort.list API endpoint and are visible as a possible filter.

- The user.get CCLE program API endpoint will return a 503 internal server error.

- **June 14, 2017**: v3.1

  Please Note:

  NOTE 1: A number of TCGA and CCLE case IDs shown below will have been removed from all cohorts since they are no longer available from NCI's Genomics Data Commons, and ISB-CGC is trying to mirror that data as closely as possible.

  TCGA cases: TCGA-33-4579, TCGA-35-3621, TCGA-66-2746, TCGA-66-2747, TCGA-66-2750, TCGA-66-2751, TCGA-66-2752, TCGA-AN-A0FE, TCGA-AN-A0FG, TCGA-BH-A0B2, TCGA-BR-4186, TCGA-BR-4190, TCGA-BR-4194, TCGA-BR-4195, TCGA-BR-4196, TCGA-BR-4197, TCGA-BR-4199, TCGA-BR-4200, TCGA-BR-4205, TCGA-BR-4259, TCGA-BR-4260, TCGA-BR-4261, TCGA-BR-4263, TCGA-BR-4264, TCGA-BR-4265, TCGA-BR-4266, TCGA-BR-4270, TCGA-BR-4271, TCGA-BR-4272, TCGA-BR-4273, TCGA-BR-4274, TCGA-BR-4276,

TCGA-BR-4277, TCGA-BR-4278, TCGA-BR-4281, TCGA-BR-4282, TCGA-BR-4283, TCGA-BR-4284, TCGA-BR-4285, TCGA-BR-4286, TCGA-BR-4288, TCGA-BR-4291, TCGA-BR-4298, TCGA-BR-4375, TCGA-BR-4376, TCGA-DM-A286, TCGA-E2-A1IP, TCGA-F4-6857, TCGA-GN-A261, TCGA-O2-A5IC, TCGA-PN-A8M9

CCLE cases: LS123, LS1034

NOTE 2: The number of cases and samples when viewed in the User Interface as compared to the BigQuery tables vary across all three projects (TCGA, TARGET, and CCLE). This is because the user interface reflects the data available at the Genomic Data Commons, whereas data in BigQuery reflects either data at the original TCGA data coordinating center supplemented with Genomic Data Commons Data (for TCGA and CCLE), or for TARGET, data received from the TARGET data coordinating center, not the Genomic Data Commons.

NOTE 3: We have removed Google Genomics functionality from the user interface. You will still be able to access CCLE open access data in Google Genomics from the command line. We are open to adding Google Genomics controlled data back into the user interface if you have a use case for it. Also we are restructuring the handling of multiple Programs of data. Please feel free to provide feedback.

NOTE 4: For TARGET data the clinical and Gene Expression files themselves are available in the system. The bam files will be available soon!

Known Issues in Sprint 16 as of 06/14/2017

- Analysis Type : Seq peek Formatting Elongated on occasion

- The CCLE data in the Webapp is not exactly the same as the CCLE data in BigQuery.

- Users cannot plot any data from a CCLE cohort on a worksheet.

- In the Webapp, the log scale on graphs does not function properly for duplicated worksheets.

- On the existing cohorts table list page, the confirmation delete 'blue x' button does not remove selected cohort if you select another option e.g Set Operation. The same issue can be found in reverse if you select the 'blue x' on the confirmation page for set operation you can then select the delete button and see the cohort on the confirmation panel.

- Swap values is not working properly for the plot settings.

- The set operation for existing cohorts complement is behaving exceptionally slow.

- A duplication of the exact cohort happens when you select the confirmation multiple times while the page is loading working with Set Operations.

- When working with a new worksheet or a duplicate worksheet with workbooks for categorical features e.g bar chart you can select the log option. The log option only applies to numerical options.

- When working with workbooks, if you select the delete confirmation button multiple times while the page is loading you will be sent to an error page.

- You currently cannot plot user uploaded data when working with workbooks.

- When plotting, certain values will be displayed as numerical when it should be a categorical value e.g Tobacco Smoking History.

- On the IGV browser when working with TCGA data build hg38 you get a No feature found with name "efgr" at the bottom of the iGV browser page.

- On the cohort creation page for TCGA data the filters disease code and project short name NA is an option which is not a valid disease.

- The mouse over feature is currently disabled for program TARGET with disease code ALL.

– The sharing feature for Workbooks, Cohorts, and User Uploaded Programs is currently disabled.

Issues that have been resolved in sprint 16 as of 06/14/2017

New Enhancements

– You will be returned a more detailed error message when uploading your own user data.

– On the Data Availability section on the cohort details page now displays the HG38 somatic mutation information for program TCGA.

Bug Fixes

– There is now a 2000 character limit for the workbook title section.

– When selecting the cohort link to complete analysis section on a worksheet will send you to the existing cohort list table page.

– Latency issues when working with the cohort creation page have been resolved.

– When working with TCGA data the IGV browser will not give you a 401 or a 404 error.

– The mouse over feature will display the long name for disease code and project short name for all programs.

– On the cohort creation page you can now filter with the HG38 somatic mutation data by gene for program TCGA using the Molecular tab.

– On the IGV Browser when working with TCGA genomic build hg38 you will no longer get a 404 error.

– On the cohort creation page when working with User Data tab, the left filter panel sorts the other filter.

– Cohorts created with API specific filters are now accessible to access by their cohort details page.

– You are now able to plot miRNA data with genomic build hg38 for TARGET data.

• **May 25, 2017: v3.0** In collaboration with the GDC we now have TARGET pediatric cancer data available for analysis in the user interface. You are now able to create cohorts and plot analysis with information from TARGET, TCGA, and CCLE data.

In addition, we have replaced the previous APIs with a new version that supports the new user interface.

We have also released the analyzed data types that are based on genome build GRCh38 for TCGA and TARGET data. GRCh37 (HG19) is also still available for TCGA, TARGET, and CCLE datasets.

Workbooks, cohorts, and variables favorites list created before the data structure migration will still be available for analysis and have been labeled as legacy and version 1. If you have difficulty using version 1 workbooks, please contact us

Please Note:

NOTE 1:A number of TCGA and CCLE case IDs shown below will have been removed from all cohorts since they are no longer available from NCI's Genomics Data Commons, and ISB-CGC is trying to mirror that data as much as possible.

TCGA cases: TCGA-33-4579, TCGA-35-3621, TCGA-66-2746, TCGA-66-2747, TCGA-66-2750, TCGA-66-2751, TCGA-66-2752, TCGA-AN-A0FE, TCGA-AN-A0FG, TCGA-BH-A0B2, TCGA-BR-4186, TCGA-BR-4190, TCGA-BR-4194, TCGA-BR-4195, TCGA-BR-4196, TCGA-BR-4197, TCGA-BR-4199, TCGA-BR-4200, TCGA-BR-4205, TCGA-BR-4259, TCGA-BR-4260, TCGA-BR-4261, TCGA-BR-4263, TCGA-BR-4264, TCGA-BR-4265, TCGA-BR-4266, TCGA-BR-4270, TCGA-BR-4271, TCGA-BR-4272, TCGA-BR-4273, TCGA-BR-4274, TCGA-BR-4276, TCGA-BR-4277, TCGA-BR-4278, TCGA-BR-4281, TCGA-BR-4282, TCGA-BR-4283, TCGA-BR-4284, TCGA-BR-4285,

TCGA-BR-4286, TCGA-BR-4288, TCGA-BR-4291, TCGA-BR-4298, TCGA-BR-4375, TCGA-BR-4376, TCGA-DM-A286, TCGA-E2-A1IP, TCGA-F4-6857, TCGA-GN-A261, TCGA-O2-A5IC, TCGA-PN-A8M9

CCLE cases: LS123, LS1034

NOTE 2: The number of cases and samples when viewed in the User Interface as compared to the Big-Query tables vary across all three projects (TCGA, TARGET, and CCLE). This is because the user interface reflects the data available at the Genomic Data Commons, whereas data in BigQuery reflects either (for TCGA and CCLE) data at the original TCGA data coordinating center supplemented with Genomic Data Commons Data, or for TARGET, data received from the TARGET data coordinating center, not the Genomic Data Commons.

NOTE 3: We have removed Google Genomics functionality from the user interface. You will still be able to access CCLE open access data in Google Genomics from the command line. We are open to adding Google Genomics controlled data back into the user interface if you have a use case for it. Also we are restructuring the handling of multiple Programs of data. Please feel free to provide feedback.

NOTE 4: For TARGET data the clinical and Gene Expression files themselves are available in the system. The bam files will be available soon!

Known Issues in this Data Structure Migration Sprint as of 05/25/2017

– Analysis Type : Seq peek Formatting Elongated on occasion

– The CCLE data in GUI is not parallel to the CCLE data in BigQuery.

– If the user shares a cohort neither the owner nor the person who was granted access to cohort will receive a confirmation email.

– Cannot plot any data if you use a CCLE data cohort on a worksheet.

– When a user duplicates a worksheet, then tries to implement the log scale it will not function properly.

– On the existing cohorts table list page, the confirmation delete 'blue x' button does not remove selected cohort if you select another option e.g Set Operation. The same issue can be found in reverse if you select the 'blue x' on the confirmation page for set operation you can then select the delete button and see the cohort on the confirmation panel.

– On the cohort view files page there are capitalization bugs on the Platform filter.

– Swap values is not working properly for the plot settings.

– The set operation for existing cohorts complement is behaving exceptionally slow.

– A duplication of the exact cohort happens when you select the confirmation multiple times while the page is loading working with Set Operations.

– When working with a new worksheet or a duplicate worksheet with workbooks for categorical features e.g bar chart you can select the log option. The log option only applies to numerical options.

– When working with workbooks, if you select the delete confirmation button multiple times while the page is loading you will be sent to an error page.

– When working on a scatter plot the Tobacco Smoking being used as the Legend is displayed in numerical values when it should be displayed as categorical values.

– The character limit for a workbook title name is currently inactive, if you exceed the possible limit you will be sent to an error page.

– You currently cannot plot user uploaded data when working with workbooks.

– Selecting cohort from worksheet "To Complete Analysis" section will send you to a 400 Bad Request error.

- You will experience latency issues when working with the create a new cohort page.

- When plotting, certain values will be displayed as numerical when it should be a categorical value e.g Tobacco Smoking History.

- The Data File Availability Panel for program CCLE in currently inactive when on the cohort details page and also editing a cohort with CCLE data.

- On the File List page you currently unable to access the bam files for the IGV Browser associated to build hg38 when working with TCGA data.

Issues that are resolved in the data structure migration sprint as of 05/25/2017

New Enhancements

- You will be returned a more detailed error message when uploading your own user data.

- The user interface now displays the same nomenclature as the Genomic Data Commons (GDC).

Bug Fixes

- The user data upload is enabled and users can now upload their own datasets and create cohorts using existing programs and newly uploaded data by the user.

- You can now have multiple Google Cloud Projects associated to your account and use only one bucket and dataset on one project with no interference.

- **April 12, 2017:** Please Note: We are currently having issues viewing bam files using the IGV browser for TCGA and CCLE data. We are working to fix the issue and it should be resolved as soon as possible.

- **February 26, 2017:**

  NOTE 1: We have removed Google Genomics functionality from the user interface. You will still be able to access CCLE open access data in Google Genomics from the command line. We are open to adding Google Genomics controlled data back into the user interface if you have a use case for it. Also we are restructuring the handling of multiple Programs of data. Please feel free to provide feedback.

  NOTE 2: There will be a reduced number of releases and features over the next month (or so) while we do some rework required for enabling the distribution of additional data sets and types copied from the NCI-GDC. The new data type is TARGET data, and different analyzed data types are based on the hg38 genome builds. Stay tuned in likely the early part of 2017.

  NOTE 3: User data uploads are currently disabled. Any projects you have previously uploaded will continue to be available in your Saved Projects list, and you can continue to work with them, but new data cannot be added at this time. We are working on bringing this function up again, please stay tuned.

Known issues in Sprint 15 as of 02/26/2017

- Analysis Type : Seq peek Formatting Elongated

- The CCLE data in GUI is not parallel to the CCLE data in BigQuery.

- If the user shares a cohort neither the owner nor the person who was granted access to cohort will receive a confirmation email.

- Cannot plot any data if you use a CCLE data cohort on a worksheet.

- When a user duplicates a worksheet, then tries to implement the log scale it will not function properly.

- On the existing cohorts table list page, the confirmation delete 'blue x' button does not remove selected cohort if you select another option e.g Set Operation. The same issue can be found in

reverse if you select the 'blue x' on the confirmation page for set operation you can then select the delete button and see the cohort on the confirmation panel.

– On the cohort view files page there are capitalization bugs on the Platform filter.

– Swap values is not working properly for the plot settings.

– The set operation for existing cohorts complement is behaving exceptionally slow.

– A duplication of the exact cohort happens when you select the confirmation multiple times while the page is loading working with Set Operations.

– When working with a new worksheet or a duplicate worksheet with workbooks for categorical features e.g bar chart you can select the log option. The log option only applies to numerical options.

– If multiple Google Cloud Projects are registered through the user interface, it is advised to to add Google buckets and BigQuery datasets to both projects currently.

– When working with workbooks, if you select the delete confirmation button multiple times while the page is loading you will be sent to an error page.

– When working on a scatter plot the Tobacco Smoking being used as the Legend is displayed in numerical values when it should be displayed as categorical values.

– The character limit for a workbook title name is currently inactive, if you exceed the possible limit you will be sent to an error page.

Issues that are resolved in Sprint 15 as of 02/26/2017

Bug Fixes

– User will no longer be sent to the Social Network Login page when trying to login. If this occurs, please feel free to send ISB-CGC feedback using this link feedback.

• **November 30, 2016**: v1.13

NOTE 1: We have removed Google Genomics functionality from the user interface. You will still be able to access CCLE open access data in Google Genomics from the command line. We are open to adding Google Genomics controlled data back into the user interface if you have a use case for it. Also we are restructuring the handling of multiple Programs of data. Please feel free to provide here.

NOTE 2: There will be a reduced number of releases and features over the next month (or so) while we do some rework required for enabling the distribution of additional data sets and types copied from the NCI-GDC. The new data type is TARGET data, and different analyzed data types are based on the hg38 genome builds. Stay tuned in likely the early part of 2017.

Known issues in Sprint 14 as of 11/30/2016

– Analysis Type : Seq peek Formatting Elongated

– The CCLE data in GUI is not parallel to the CCLE data in BigQuery.

– User will occasionally be sent to the Social Network Login page when trying to login. If this occurs, please go the the home page of the Web Application and try again.

– If the user shares a cohort they do not receive a confirmation email.

– Cannot plot any data if you use CCLE data cohort on a worksheet.

– When a user duplicates a worksheet, then tries to implement the log scale it will not function properly.

– If a researcher leaves the workbooks inactive the page freezes.

- On the existing cohort list page for the delete button, select the blue x does nothing. It should be disabled.

- On the cohort view files page there are capitalization bugs on the Platform filter.

- Swap values is not working properly for the plot settings.

- Some plot setting are saved or retrieved when working with worksheets.

- The set operation for existing cohorts intersection is behaving exceptionally slow.

Issues that are resolved in Sprint 14 as of 11/30/2016

Bug Fixes

- The user can no longer see BCGSC expression as an option when plotting genes if user does not select center filter on worksheet.

- Worksheets added to an existing workbook now behave the same as the original worksheet.

- Cohort set operations no longer performing exceptionally slow.

- **November 16, 2016**: v1.12

  Please Note: We are removing Google Genomics from the user interface. You will still be able to access CCLE open access data in Google Genomics from the command line. We are open to adding Google Genomics controlled data back into the user interface if you have a use case for it. Please feel free to provide feedback.

  Known issues in Sprint 13 as of 11/16/2016

  - Analysis Type : Seq peek Formatting is Elongated

  - The CCLE data in GUI is not parallel to the CCLE data in BigQuery.

  - User will occasionally be sent to the Social Network Login page when trying to login. If this occurs, please go the the home page of the Web Application and try again.

  - If the user shares a cohort they do not receive a confirmation email.

  - Cannot plot any data if you use CCLE data cohort on a worksheet.

  - When a user duplicates a worksheet, then tries to implement the log scale it will not function properly.

  - If a researcher leaves the workbooks inactive the page freezes.

  - On the existing cohort list page for the delete button, selecting the blue x does nothing. It will be be disabled in a future release.

  - On the cohort view files page there are capitalization bugs on the Platform filter.

  - Swap values is not working properly for the plot settings.

  - Some plot setting are saved or retrieved when working with worksheets.

  - Worksheets added to an existing workbook behave differently than the original worksheet.

  - The user can see BCGSC expression as an option when plotting genes if user does not select center filter on worksheet.

  - The set operation for existing cohorts intersection is behaving exceptionally slow.

  Issues that are resolved in Sprint 13 as of 11/16/2016

  New Enhancements

– A warning will be displayed if the user is trying to plot with required data missing e.g. must select an analysis, gene or variable, and a cohort to create a plot.

– On the project details page user will be sent to upload new study in existing project tab when they select upload data.

– When the user plots a graph with NA values, you will be returned a notification stating no valid data was found.

– There is no longer text overlapping on the Cloud Hosted Datasets readthedocs page in the documentation.

Bug Fixes

– The user can no longer add the same gene symbol twice if list to the same worksheet even if they have given their list different names.

– When the user selects multiple cohorts for color by feature for scatter plot all cohorts selected display on the graph.

– On the existing cohorts table for public cohorts, the new workbook and set operations buttons are now active.

– For all analysis types the x-axis and y-axis with certain variables text will no longer overlap and is displayed clearly.

– The upload data button is disabled on the review files page when no buckets or datasets are associated.

– Someone with multiple eRA accounts will be no longer have issues when trying to access controlled data.

- **November 2, 2016**: v1.11

  Known issues in Sprint 12 as of 11/02/2016

  – The user can add same gene twice if list to the same worksheet it they have different names.

  – Analysis Type : Seq peek Formatting Elongated

  – The CCLE data in GUI is not parallel to the CCLE data in BigQuery.

  – If a user creates a cohort with sample type filter Cell Lines and CCLE the total number of samples count off by one.

  – User will occasionally be sent to the Social Network Login page when trying to login. If this occurs, please go the the home page of the Web Application and try again.

  – If the user shares a cohort they do not receive a confirmation email.

  – When the user selects multiple cohorts for color by feature for scatter plot they do not display in chart.

  – Cannot plot any data if you use CCLE data cohort on a worksheet.

  – When the user plots a graph with NA values the UI returns a blank graph.

  – When a user duplicates a worksheet, then tries to implement the log scale it will not function properly.

  – If a researcher leaves the workbooks inactive the page freezes.

  – On the existing cohort list page for the delete button, selecting the blue x does nothing. It should be disabled.

  – On the cohort view files page capitalization bugs on the Platform filter.

– Swap values is not working properly for the plot settings.

– Some plot settings are saved or retrieved when working with worksheets.

– On the existing cohorts table for public cohorts, the new workbook and set operations buttons are currently inactive.

– Worksheets added to an existing workbook behave differently than the original worksheet.

Issues that are resolved in Sprint 12 as of 11/02/2016

New Enhancements

– Introduce user data upload functionality see documentation here.

– More fluid zoom feature when working with analysis worksheets.

– Case Sensitivity is now maintained in creating and displaying Workbook names throughout the entire User Interface.

– You can now create a new cohort from the menu bar.

– Variables menu bar is displayed similar to the rest of the favorites variables.

– On the dashboard, all create new buttons/links are identical.

– Owner of what is shared either a workbook or a cohort is able to remove multiple viewers. Viewers are also able to remove themselves.

– Removed BCGSC gene expression from the UI gene specification selection for plot analysis.

Bug Fixes

– X or Y- Axis for text no longer overlaps on worksheet for any analysis type, except for violin plot.

– The Legend is no longer displayed elongated when you use multiple cohort for color by feature for violin plot.

– miRNA_expression_values_fixed table in dataset 2016_07_09_tcga_data_open reflect only hg19.mirbase20 files.

– You are now able to duplicate a workbook that has been shared with you by someone else.

– Added pseudo-counts to the mosaic plots on the create new cohort page. This allows you to be sure of always being able to see (and select) the smallest contributors in these mosaics.

– Removing the filter from the filter confirmation from the create new cohort page, this will remove it from the rest of filter selections.

– Select the "check-all" feature on the create new cohort page will no longer cause duplicates on the selected filters panel.

– Create cohort from plot selection now works with all analysis types.

– Data inconsistencies between the create new cohort histogram filter and the most recent Big-Query datasets has been addressed and resolved.

• **September 21, 2016**: v1.10

Known issues in Sprint 11 as of 9/21/2016

– The user can add same gene twice if list to the same worksheet it they have different names.

– The Bar chart on the worksheet panel renders overlapping text.

– Analysis Type : Seq peek Formatting Elongated

– The CCLE data in GUI is not parallel to the CCLE data in BigQuery.

– If a user creates a cohort with sample type filter Cell Lines and CCLE the total number of samples count off by one.

– User will occasionally be sent to the Social Network Login page when trying to login. If this occurs, please go the the home page of the Web Application and try again.

– If the user shares a cohort they do not receive a confirmation email.

– The Legend is displayed elongated when you use multiple cohort for color by feature for violin plot.

– When the user selects multiple cohorts for color by feature for scatter plot they do not display in chart.

– Cannot plot any data if you use CCLE data cohort on a worksheet.

– When the user plots a graph with NA values the UI returns a blank graph.

– When a user duplicates a worksheet, then tries to implement the log scale it will not function properly.

– There are duplicate rows in the molecular data table in BigQuery.

Issues that are resolved in Sprint 11 as of 9/21/2016

New Enhancements

– Text in confirmation box of a duplication of a workbook has been enhanced.

– On the registered Google Cloud Projects page, icon has been added for the user to go directly to the Google Cloud Console page if desired.

– When the a Service Account is removed from the Access Control List, the project owner is sent an email with an explanation as to why the account was removed.

– IGV File List page displays of which page user is browsing.

Bug Fixes

– For a Cubby hole plot the x - axis name can be seen clearly.

– On a duplicate worksheet when working with gene specifications, user is able to select between all options multiple times.

– Page becomes elongated when the user builds a Cubby Hole plot.

– The selected variables for the plot setting on a worksheet are saved after the user leaves the workbook.

– When registering a Google Cloud Project the user is displayed the list of emails associated to the GCP only once.

• **September 7, 2016**: v1.9

Known issues in Sprint 10 as of 9/07/2016

– The user can add same gene twice if list to the same worksheet it they have different names.

– The Bar chart on the worksheet panel renders overlapping text.

– Analysis Type : Seq peek Formatting Elongated

– The CCLE data in GUI is not parallel to the CCLE data in BigQuery.

– If a user creates a cohort with sample type filter Cell Lines and CCLE the total number of samples count off by one.

- – User will occasionally be sent to the Social Network Login page when trying to login. If this occurs, please go the the home page of the Web Application and try again.

- – Page becomes elongated when the user builds a Cubby Hole plot.

- – X-axis name cut off for cubby hole plot when x-axis has only 3 criteria.

- – If the user shares a cohort they do not receive a confirmation email.

- – The Legend is displayed elongated when you use multiple cohort for color by feature for violin plot.

- – When the user selects multiple cohorts for color by feature for scatter plot they do not display in chart.

- – When the user creates a duplicate worksheet,the bar chart with a gene with specification protein can freeze when selecting an option for the Select Feature.

- – Cannot plot any data if you use CCLE data cohort on a worksheet.

- – When the user plots a graph with NA values the UI returns a blank graph.

- – When a user duplicates a worksheet, some functionality related to plotting will not function properly on the duplicate worksheet.

Issues that are resolved in Sprint 10 as of 9/07/2016

New Enhancements

- – Dictionary mapping feature types to units for use in plot displays added to worksheets.

- – The user now has the option to make the axis logarithmic if the plot can display continuous numerical data for eg. mRNA expression levels.

- – The NIH username entry is now case insensitive for dbGaP authorization.

- – The mouse over feature works when the user has created a long workbook name on the existing workbooks table page.

- – The mouse over functionality was added to the worksheet name within a workbook.

Bug Fixes

- – The order by ascending or descending feature is now working properly for the existing workbooks table page.

- – Tobacco Smoking History filter in the create cohort page displays the filters in descriptive values.

- – The user can now select all existing cohorts when on the add cohort(s) to worksheet page.

- – The gene specification selection on the worksheet page is now working properly.

- – When a user shares a workbook with someone the person who received viewer access to the workbook is sent a confirmation email. If the person who shared the workbook then deletes the workbook before it's opened, then the person clicks the invitation link the person is sent to the unknown invitation page. The button to go back to the Dashboard page appears like this, "Your Dashboard"

- – The user is sent an email when the Service Account is removed the Access controlled list for having a user associated to the project who is not dbGaP authorized.

- **August 24, 2016**: v1.8

  Known issues in Sprint 9 as of 8/24/2016

  - – The user can add same gene twice if list to the same worksheet it they have different names.

- The Bar chart on the worksheet panel renders overlapping text.

- Analysis Type : Seq peek Formatting Elongated.

- The CCLE data in GUI is not parallel to the CCLE data in BigQuery.

- If a user creates a cohort with sample type filter Cell Lines and CCLE the total number of samples count off by one.

- User will occasionally be sent to the Social Network Login page when trying to login. If this occurs, please go the the home page of the Web Application and try again.

- Page becomes elongated when the user builds a Cubby Hole plot.

- X-axis name cut off for cubby hole plot when x-axis has only 3 criteria.

- When the user shares a cohort they do not receive a confirmation email.

- User will be spammed with email every one minute when their service account is removed from the ACL control list. To stop this, please either delete your service account from the ISB-CGC interface, or remove the GCP project member(s) who is (are) not authorized to access the controlled data set. (see documentation here). We are planning to reduce the frequency of the notification emails to once per day.

- The Legend is displayed elongated when you use multiple cohort for color by feature for violin plot.

- When the user selects multiple cohorts for color by feature for scatter plot they do not display in chart.

- When the user creates a duplicate worksheet,the bar chart with a gene with specification protein can freeze when selecting an option for the Select Feature.

- When a user shares a workbook with someone the person who received viewer access to the workbook is sent a confirmation email. If the person who shared the workbook then deletes the workbook before it's opened, then the person clicks the invitation link the person is sent to the unknown invitation page. The button to go back to the Dashboard page appears like this, "Your Dashboard{"

- Cannot plot any data if you use CCLE data cohort on a worksheet.

Issues that are resolved in Sprint 9 as of 8/24/2016

New Enhancements

- When the researcher is on the Register Service Account page, after they have submitted the Service Account associated to their Google Cloud Project a table that shows who is authorized will be prompted.

- There is now a column that says "Has NIH Identity", before it said, "Has eRA Commons".

- When the researcher creates a new cohort with more than 20 filters chosen the URL exceeds the limit of 2K characters and this affects the count for the Details panel. Therefore the user is now prompted with an alert box that will say, "You have selected too many filters. The current counts shown will not be accurate until one or more filter options are removed." if this is ever the case.

- In the user details page, if the researcher has not registered a Google Cloud Project it will say, "Register a Google Cloud Project" on the link.

Bug Fixes

- The researcher can now delete whom they share cohort with from existing cohorts table.

- After 24-hours of use, a dbGaP authorized user can re-authenticate through the link provided in the user details page.

- The variable favorites list table page can now support a long title for the variable list.

- The filter name will appear aligned in the verification panel when the filter is name too long for the create in cohort filter confirmation selection on the create new cohort page.

- Grouped Data Type filter counts (Methylation, RNA Seq, miRNA Seq) now behave like the other count groups. The counts will behave as grouped values.

- The user can no longer select a categorical variable for selection for Histogram plot.

- The Filter token displays are now shown in 'readable' names when working with cohort filters.

- Controlled access BAM files are now viewable viewable in the IGV browser after the user has authorized their credentials.

- The user can now unlink an eRA commons account from their Google Identity in the user detail page.

- The violin plot was inconsistently failing. We have updated the JavaScript, therefore the Violin plot no longer fail.

- **August 10, 2016**: v1.7

  New Functionality Released in this Sprint

  - The researcher can now create a cohort of participants and samples based on the presence of a gene mutation in a specified gene. Look for the new "Molecular" tab when you are creating a cohort.

  - The bioinformatics programmer now has the ability to associate their Google Cloud Project's Service Account. This allows the researcher to run computational pipelines from Google Virtual Machines using TCGA Controlled data (e.g. BAM files) for seven days before they have to reauthorize. For more information please select here.

  Known issues in Sprint 8

  - The user can add same gene twice if list to the same worksheet it they have different names.

  - The Bar chart on the worksheet panel renders overlapping text.

  - Cannot delete whom you share cohort with from existing cohorts table.

  - Analysis Type : Seq peek Formatting Elongated

  - The CCLE data in GUI is not exactly coordinated the CCLE data in BigQuery.

  - If a user creates a cohort with sample type filter Cell Lines and CCLE the total number of samples count is off by one.

  - After 24-hours of use, a dbGaP authorized user has to logout and then log back in to be prompted with NIH login link to re-access controlled data.

  - User will occasionally be sent to the Social Network Login page when trying to login. If this occurs, please go the the home page of the Web Application and try again.

  - Page becomes elongated when the user builds a Cubby Hole plot.

  - X-axis name cut off for Cubby Hole plot when x-axis has only 3 criteria.

  - When the user shares a cohort they do not receive a confirmation email.

  - When a name is too long for variable favorites list table, the Last Updated" column will appear cut off.

  - Filter name will appear off the verification panel when the filter is name too long for the create in cohort filter selection.

- Grouped Data Type filter counts (Methylation, RNA Seq, miRNA Seq) don't behave like other count groups. The counts behave as though the values were for distinct categories.

- User will be spammed with email every one minute when their service account is removed from the ACL control list. To stop this, please either delete your service account from the ISB-CGC interface, or remove the GCP project member(s) who is (are) not authorized to access the controlled data set. (see documentation here). We are planning to reduce the frequency of the notification emails to once per day.

- The user can select a categorical variable for selection for Histogram plot, and will return a graph with no data.

- The Legend is displayed elongated when you use multiple cohort for color by feature for violin plot.

- When the user selects multiple cohorts for color by feature for scatter plot they do not display in chart.

- When the user creates a duplicate worksheet,the bar chart with a gene with specification protein can freeze when selecting an option for the Select Feature.

Issues resolved in Sprint 8

New Enhancements

- The user now has the option to select all or deselect all possible filters for any tab that has more than 10 possible options in the create new cohort page.

- The user can now set all existing tables by either ascending or descending order.

- The cohort_id has been added to the detail cohort page. This allows the user to reference a desired cohort with ease in the API endpoints.

- When creating a new cohort, the user is given the full description for sample type in the selected filters panel.

Bug Fixes

- Histological Type entries in create new cohort page on the user interface now match the Google BigQuery entries in terms of capitalization.

- Filters for data type counts in left panel currently is now working properly.

- When a user sets a cohort as Color by feature for violin plot legend will be set to cohort. Then when the user sets another color by feature it will update the legend.

- The user can no longer make a gene list without selecting a gene first.

- The user can now list the Last Modified section for the existing cohort table by either ascending or descending order.

- In the create new cohort page for the data type tab, the user can now select either True or False for DNA Sequencing, Protein, and SNP Copy Number filters.

- When the user edits a new cohort and sets the edited cohort to return zero samples, the user will be prompted to select different set of filters.

- **July 20, 2016**: v1.6

  Known issues in Sprint 7

  - The user can add same gene twice if two identical worksheets with different names are uploaded.

  - The Bar chart on the worksheet panel renders overlapping text.

  - User cannot delete whom you share cohort with from existing cohorts table.

– Analysis Type : Seq peek Formatting Elongated.

– The CCLE data in GUI is not parallel to the CCLE data in BigQuery.

– If a user creates a cohort with sample type filter Cell Lines and CCLE the total number of samples count off by one.

– Histological Type entries in create new cohort page on the user interface should match the Google BigQuery entries in terms of capitalization.

– When a user sets a cohort as Color by feature for violin plot legend will remain cohort.

– After 24 hour dbGaP authorization runs out the user is unable to re authenticate. (If you have this issue, please log out and log back in to be prompted with login link for dbGaP authorization.)

Issues resolved in Sprint 7

New Enhancements

– Created ability in GUI to make cohorts based on presence of an HPV status.

– Created ability in GUI to make cohorts based on BMI value.

– In the details panel for existing cohort have a section that shows the ISB-CGC cohort_id.

– Enhancements of GUI to view submenu item in different screen sizes and resolutions.

– New version of IGV javascript installed.

Bug Fixes

– User can no longer add same filter to existing cohorts.

– Optimized Security in the user interface.

– If a user opens a shared cohort it will appear once on the dashboard.

– Pathologic State Filter in create cohort Stage is displayed capitalized.

– Filter counts with 0 value do list when editing a pre-existing cohort.

– Filters for data type counting in left panel is working properly.

– After 24 hour dbGaP authorization runs out the user is able to re authenticate.

– User can not create new gene list without giving the gene list a name.

• **July 6, 2016**: v1.5

Known issues in Sprint 6

– The user can add same gene twice if list to the same worksheet it they have different names.

– The user can add same filter to existing cohorts.

– The Bar chart on the worksheet panel renders overlapping text.

– Cannot delete whom you share cohort with from existing cohorts table.

– Analysis Type : Seqpeek Formatting Elongated.

– The CCLE data in GUI is not parallel to the CCLE data in BigQuery.

– If a user opens a shared cohort it will appear twice on the dashboard.

– If a user creates a cohort with sample type filter Cell Lines and CCLE the total number of samples count are off by one.

– Pathologic State Filter in create cohort Stage should be displayed capitalized.

- Histological Type entries in create new cohort page on the user interface should match the Google BigQuery entries in terms of capitalization.

- Filter counts with 0 value don't list when editing a pre-existing cohort.

- Filters for data type counting in left panel currently is not working properly.

Issues resolved in Sprint 6

New Enhancements

- A user can only select the cloud storage checkbox if he or she has been authenticated and authorized through the user details page. Otherwise the user can view the cloud storage checkbox but there will be a disabled cursor icon when the user hovers over in an attempt to select the checkbox.

- The counts for the queries were refactored to match what was done for the APIs .

- The Download File List as CSV was refactored to a maximum of 65,000 files at once.

- Date formats on Workbooks, Cohort, Gene, and Variables list pages all reflect the same format.

- The Last Updated columns to variable and gene lists were added to the user Dashboard

Bug Fixes

- The user can now select a cohort in the color by feature section for the violin and the scatter plots in the worksheet section.

- The Gene list variable used for analysis in the worksheet plot settings section is the exact gene as compared to a gene that contains the string.

- The Comments button for both the workbook and the cohort section, when the user clicks the request multiple times within one second the user interface will not post duplicate comments in the comments section.

- The user can now select gene HP in Create Gene list favorite page to be used for analysis. For worksheet analysis the user now has ability to select different genes once one already selected and utilized for analysis.

- In the variable favorites table, the menu for a specific variable will no longer be cut off once a certain set of variables list are exceeded.

- A 400 Error pop up window will no longer appear as the user transitions from the File List page to IGV browser page.

- The Public Data Availability section will no longer display any cut off if the user drags data type to the left of the page away from the panel itself, in detail page of existing cohort or the create new cohort page.

- When the user edits a cohort, details section will display which filter(s) were applied for each update.

- Cloud storage path in CSV file download for GA/BCGSC and GA/UNC V2 platforms can now be viewed.

- The menu bar will display existing list for variable favorites list, gene favorites list, cohorts, and workbooks with no cut off.

- When the user has selected a variable for the y-axis, the chart will display the selected variable in the charts.

- When the user clicks Save Changes when modifying an existing cohort the user can will no longer be spammed with multiple cohorts created at once when clicking the button multiple times within one second.

- The Save cohort Endpoint default example for v1 now works properly.

- For the cohort_list API endpoint v1 will now pull only the cohort_id you specified.

- **June 8, 2016**: v1.4

  Known issues in Sprint 5

  - The user can add same gene twice if list has different names.

  - The user can add same filter to existing cohorts.

  - In the Create new Cohort page, the left filters (#) does not re-populate as you select filters to match the sample number in clinical feature panel.

  - The bar chart renders overlapping text in the x-axis and y-axis for certain variables.

  - A user cannot delete whom you share a cohort with from the existing cohorts table.

  - On a worksheet with the Analysis Type : Seq peek, the formatting will display Elongated when the user selects a certain gene.

  - CCLE data in GUI is currently not parallel the CCLE data in BigQuery.

  - User currently cannot select a cohort in the color by feature section in a worksheet.

  - The Gene list used for analysis currently uses genes similar as to original gene and well as the specific gene added to list, in the plot settings menu.

  - The comments button for both workbooks/cohorts, if user clicks the comment button multiple times within one second will post duplicate comment.

  - User currently cannot select gene HP or gene's with only two letters in the Create Gene list favorite page.

  - In Violin plot - the user has no ability to select a different gene once one is already selected.

  - In the variable favorites table, the menu for a specific variable will be cut off once a certain set of variables list are exceeded.

  - A 400 Error pop up window will appear as the user transitions from the File List page to IGV browser page.

  - Public Data Availability section will be cut is user drags data type title to the left of the page away from the panel itself,in detail page of existing cohort.

  Issues resolved in Sprint 5

  New Enhancements

  - Upgraded system from using Django 1.8 to Django 1.9.

  - A link to the google cloud platform has been added to the user details page.

  - The TCGA filter is selected as the default project when creating a new cohort.

  - When the user clicks on the browser back button, the user will remain on the same worksheet that they were previously on.

  - When the user goes adds a new gene list, variable favorites list, and/or cohort from the worksheet data type panel, the button will display "Apply to Worksheet".

  - The feedback/help section has been moved to the top of the page to provide the user a more convenient way to send us feedback.

  Bug Fixes

  - User can no longer add a duplicate gene to same gene favorites list.

– To edit a gene name the user must now delete and re-type the desired gene name.

– The functionality of a duplicate worksheet drop down menu reflects the same functionality of the original worksheet.

– The Last Updated section reflects any changes made to the variable list, cohort list, and gene list in their corresponding tables.

– The File list page now allows the user to add a maximum of five files to use in the IGV browser between all the pages in the file list table.

– When a user hovers over clinical feature panel for Sample Type and Tumor Tissue Type the top row when hovered over the name is displayed clearly.

– Order by Ascending/Descending is working properly for Existing Cohorts table page.

– The user is now able to plot gene's with a hyphen(-) in the gene name itself.

– The user is now able to download a maximum of 85,000 files at a time, in the File List page for a selected cohort.

• **May 10, 2016**: v1.3

Known issues in Sprint 4

– A user can add same gene twice if identical gene list have different names.

– The user can add same filter already selected to an existing cohort.

– The create new Cohort left filters number count does not re-populate as you select filters to match sample number count in clinical feature panel.

– When a Bar chart renders overlapping text is displayed on the x-axis of the plot.

– Cannot delete whom you share a cohort with from the existing cohorts table only from the details page of a cohort.

– Analysis Type : Seq peek formatting is elongated when a user selects certain gene for analysis. Using the gene TP53 can reproduce this issue.

– The CCLE data in GUI currently does not parallel the CCLE data in BigQuery.

– A user can add a duplicate gene to same gene favorites list in the create new gene list page.

– By double clicking a gene name in the create new gene list page, the gene will expand but display a blank space.

– A duplicate worksheet will display the color by feature variables twice in the drop down list.

– A user currently cannot select a cohort in the color by feature section.

– The Gene list drop down list used for analysis should be exact gene only.

– The comments button for both workbook and cohort comments section, if the user is to click comment button multiple time within one second, this action will post a duplicate comment.

– The last Update section should reflect any changes made to variable list, cohort, and gene list for their corresponding tables.

– The user cannot select the gene HP in the Create Gene list favorite page.

Issues resolved in Sprint 4

New Enhancements

– Data Use Certification Agreement link updated and the help link was removed.

– The Data Type section in the Create new Cohort page name change from MIRNA Sequencing to miRNA Sequencing and SNP CN to SNP Copy-Number.

– The number of patients is now dynamically displayed in the create new cohort page when selecting filters in the details panel.

– The number of samples is now dynamically displayed in the create new cohort page when selecting filters in the details panel.

– By default in the create new cohort page, you will have the TCGA data filter selected.

– When creating a cohort, checking feature boxes will be throttled so as to avoid miss-represented data.

– Tooltips were added to the Sample Type section in the clinical features panel.

– Minor changes were made in personal details page.

Bug Fixes

– The Clinical Features Panel in the create new cohort page will no longer display BRCA even if unselected.

– The last updated section in existing workbooks panel does update when changes are made to existing workbook.

– Set operation Union patient number is working correctly.

– Upon duplicating a cohort it will duplicate the selected filter(s) as well.

– User is able to download file list as csv for any cohort with any filter selected.

– There is no legend cut off for violin plot or any other analysis type when the color by feature is set to Prior Diagnosis or any other variable.

– When user switches gene in plot settings the feature choices for that specification will refresh.

– The variable clinical search feature works properly when the user searches for clinical variables and then are used for analysis.

• **April 27, 2016**: v1.2

Known issues in Sprint 3

– Can add same gene twice if list has different names.

– User can add same filter to existing cohorts.

– Create new Cohort left filters (#) does not re-populate as you select filters to match sample # in clinical feature panel.

– Clinical Features Panel in create new cohort page will still display BRCA even if unselected.

– Last updated section in existing workbooks panel does not update when changes are made to existing workbook.

– Bar chart renders overlapping text.

– Set operation Union patient # off by one.

– Legend Name cut off when name is too long.

– Upon duplicating a cohort it duplicates the selected filter as well.

– Cannot delete whom you share cohort with from existing cohorts table.

– Unable to down file list as csv for any other cohort only selected filter CCLE.

– Legend Cut Off for violin plot when color by feature set to Prior Diagnosis.

– When user switches gene in plot settings the feature choices for that specification disappears.

Issues resolved in Sprint 3

New Enhancements

– The comments section now has a max number of characters 1000 limit.

– Link created to Extend controlled access period to 24-hours from the moment the link is clicked.

Bug Fixes

– A user can now click new worksheet multiple times within a few seconds and only produce one sheet.

– The user must now add a new filter in an existing cohort to edit it the cohort.

– The duplicate button for an existing cohort will only make one duplicate at a time.

– Clicking 150+ selected filters will not create an error page.

– Cancel button on Create new gene list page will send you to Gene list favorites table menu.

– Violin plot : User can not add categorial value to y-axis.

– If user edits an existing cohort, the old filter(s) will not be removed.

– If a new worksheet is generated, the worksheet functionality is working properly.

– User will get the '500: There was an error while handling your request. If you are trying to access a cohort please log out - and log back in. Sorry for the inconvenience.' if the user is inactive for more in 15 minutes when trying to create/use existing cohort.

– Clinical Feature Panel is displayed properly and reacts to filters being added/removed quickly.

– The user must have text to add a comment.

– All columns in file list table will be transferred/displayed when exported as csv file.

- **April 14, 2016**: v1.1

    Known issues in Sprint 2

    – If user clicks create in new worksheet too many times within a few seconds will create duplicate worksheets

    – Can add same gene twice if list has different names

    – Apply filters button work when no filter is selected in edit cohorts page

    – If user clicks create in new cohorts too many times within a few seconds will create duplicate cohorts

    – User can add same filter to existing cohorts

    – Clicking 150+ selected filters will create error page

    – Create new Cohort left filters (#) does not re-populate as you select filters to match sample # in clinical feature panel

    – Clinical Features Panel in create new cohort page will still display BRCA even if unselected

    – Cancel button on Create new gene list page will send you to Data Source | Gene Favorites page

    – Violin plot : User can add categorial value to y-axis

– Last updated section in existing workbooks panel does not update when changes are made to existing workbook

– If user edits an existing cohort the old filter(s) will be removed

Issues resolved in Sprint 2

New Enhancements

– Tool tips added for disease code in create new cohort page

– Disease in longname in tool tips the first letter is capitalized

Bug Fixes

– The user detail page will now display the correct date

– The plot settings for a new worksheet are now working properly

– Plot settings for duplicate worksheets are now working properly

– The plot settings will now match the analysis type for existing worksheet plot

– The user can now edit existing cohort name

– Set Operations : Intersection working properly

– Set Operations : Union working properly

– Set Operations : Complement is now working properly

– User is now able to delete selected filters from selected filter panel in new cohort page using the blue X

– Editing an existing variable favorites list will display previously selected variables

– (Already in documentation) Green checkmark will appear for IGV link

– Update plot button will now work on a duplicate worksheet(can be added with 3)

– User can now delete all cohorts with the select all feature

– Fixed bugs with Data Type Create new cohort generating errors

– The user can now search for variable favorite with the miRNA feature

– The user can now search for a variable favorite through the clinical search feature

• **March 14, 2016: v1.0**

– When working with a worksheet two plots will be generated occasionally.

– Axis labels and tick values sometimes overlap and get cutoff.

– Page elongated when Cubby Hole plot generated and there are lots of values in the y axis.

• **December 23, 2015: v0.2**

– Treemap graphs in cohort details and cohort creation pages will not apply its own filters to itself. For example, if you select a study, the study treemap graph will not update.

– Cohort file list download not working.

• **December 3, 2015: v0.1**

– First tagged release of the web-app

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.6 Programmatic Access

Programmatic access to the data and metadata is provided through a combination of ISB-CGC APIs and Google APIs. The majority of the ISB-CGC data in BigQuery tables and in Google Cloud Storage is accessed directly via Google Cloud tools and interfaces. Access to ISB-CGC metadata and user-data such as cohort definitions is provided through the ISB-CGC programmatic API described below.

A growing set of tutorials and programming examples illustrating how you can work with these hosted data sets from a variety of programming environments such as Python and R, using Google Compute Engine VMs and the Google Genomics Pipelines (GGP) service are provided in our github repositories, also described below.

1.6.1 Computational System Model

There are two primary ways in which users can interact with ISB-CGC data. The first method is through graphical interfaces such as the ISB-CGC web application or the BigQuery web interface, which provides users a convenient web-based interface from which it is easy to create and visualize collections of data hosted by the ISB-CGC.

The second method is through the ISB-CGC programmatic API or through other Google Cloud APIs. The ISB-CGC API provides access to much of the same computational functionality as the web application, and the other Google APIs can be used to access the hosted data sets depending on which technology is used to host them:

- the BigQuery Web UI, Command-Line Tool, or REST API for the data stored in BigQuery tables;
- the Google Cloud Storage (GCS) JSON API or gsutil for the data stored in GCS objects; or
- the Genomics REST API for data stored in Google Genomics.

For users interested in performing custom analyses, accessing the data directly using these APIs will provide greater flexibility.

Here are instructions on how to access BigQuery from the Google Cloud Platform.

Here are instructions on how to see ISB-CGC data through the BigQuery Web UI.

Here are examples of how to query ISB-CGC data using BigQuery, including using multiple tables with Joins.

The Cloud Paradigm

In addition to hosting the TCGA data in the cloud, one of the main goals of the ISB-CGC is to "bring the computation to the data". There are many ways that this can be done using legacy tools, cloud-native tools, or a combination of the two. Regardless of the details of the particular solution, the single most important difference between the ISB-CGC computational system model and traditional HPC models is that there is no single monolithic system that is doing the computational work. Cloud-native solutions instead abstract the configuration management process from the allocation of physical hardware, making it very easy to programmatically request an arbitrary number of identical machines, which can then be easily "torn down" (and regenerated) whenever necessary. The configuration state of these machines will always be identical on startup, and can be parametrized according to your algorithm's resource needs.

One important implication to understand about this new computational paradigm is that the burden of system administration is partially shifted to the users of the cloud: researchers and developers. While numerous tools exist to help simplify these tasks, there is no IT department managing your cloud-computing. This means that researchers will need to learn a new skill-set.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.6.2 R, Python and SQL Tutorials

For ISB-CGC users who want to perform custom analyses by writing R or Python scripts, we have begun to assemble a set of examples in two public github repositories: examples-Python and examples-R. R users can work from the familiar environment of RStudio, and Python programmers can enjoy the richness available in IPython notebooks by taking advantage of the newly released Cloud Datalab

Note: That Cloud Datalab is a beta release, but Google has released documentation on how to install Cloud Datalab locally here and how to Run Cloud Datalab notebooks on Google Cloud Platform here .

These repositories contain numerous examples that will help you learn to access and analyze the TCGA data in BigQuery, as well as examples showing how to use our APIs to query the metadata and discover where to find the data that you are looking for in Google Cloud Storage.

In addition there is the Query of the Month Club where there are multiple examples of using SQL queries to analyze the data housed in BigQuery.

We encourage the community to provide feedback on these tutorials and also to add your own examples to enrich this public resource!

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.6.3 Programmatic Interfaces

```
The changes needed to support multiple programs have rendered the V1 and V2 APIs non-
↪functional
and therefore users must migrage all API calls to the V3 version.  Note that this␣
↪usually means
just a minor adjustment to the URL.  Some of the examples in the github repository
may still reference the V1 or V2 API.
```

Programmatic access to molecular data and metadata within the ISB-CGC platform uses a combination of ISB-CGC APIs and Google APIs, as illustrated by the block diagram on the front page of this documentation.

- The **ISB-CGC API** provides programmatic access to data and metadata stored in CloudSQL. This includes information describing TCGA patients and samples, data availability, user-created cohorts, *etc.* In this section of our documentation, you will find more details about using the ISB-CGC API.

- Native **Google APIs** are used for optimized, high-speed programmatic access to molecular data in BigQuery, Google Cloud Storage, or Google Genomics. Code examples illustrating usage of these Google APIs are available in the ISB-CGC code repositories on github. Additional Google Cloud Platform Documentation for some of the key technologies leveraged by the ISB-CGC platform can be found by following these links:

    - BigQuery APIs & Reference

    - Cloud Storage APIs & Reference

    - Genomics API Overview

ISB-CGC API

The **ISB-CGC API** provides an interface to the ISB-CGC metadata stored in CloudSQL, and consists of several "endpoints", implemented using Google Cloud Endpoints. Details about these endpoints can be found here, and examples illustrating usage from R and Python can be found in our examples-R and examples-Python repositories on github.

Some example use-cases include:

- obtaining a list of *patient identifiers* based on a defined set of criteria;

- obtaining a list of *sample identifiers*, associated with a specific patient;

- obtaining detailed *metadata* about a particular patient or sample;

- creating (or retrieving a previously saved) *cohort* of patients and samples, based on a defined set of criteria;

- obtaining a list of *data files* in Cloud Storage, associated with a specific *sample*, *cohort*, *platform*, or *data-type* (or any combination thereof);

The APIs Explorer can be used to see details about each endpoint, and also provides a convenient interface to test an endpoint through your web browser. Following the link in the previous sentence will take you to a page with a list of APIs, in which each API consists of a set of functionally-related endpoints. Together, these individual APIs make up the **ISB-CGC API**. (Note that not all of these APIs are intended for direct use by end-users: some are intended for use only by the ISB-CGC Web-App, as described in the information on the first APIs Explorer page mentioned above.)

Cohorts are the primary organizing principle for subsetting and working with the TCGA data. A cohort is a list of samples and a list of patients. Users may create and share cohorts using the ISB-CGC web-app and then programmatically access these cohorts using this API. (TCGA samples are identified using a 16-character "barcode" *eg* TCGA-B9-7268-01A, while patients are identified using the 12-character prefix, *ie* TCGA-B9-7268, of the sample barcode. Other datasets such as CCLE may use other less standardized naming conventions).

Usage

Endpoints are simple https GET or PUT requests, *eg*:

```
V3 TCGA - GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
↪cases/TCGA-B9-7268
V3 TARGET - GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/
↪target/cases/TARGET-20-PABLDZ
V3 CCLE - GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/
↪cases/FU-OV-1
```

The first three GET commands above illustrates the usage with the new program-specific V3 endpoints.

The url (without the "GET" command) can also be pasted directly into your browser, like this or this. Packages are available in most languages to allow you to easily perform https GET and PUT requests, such as the httr package for R, and the Python requests library.

In addition, the Google Python API Client Library can be used to build a service object which provides a functional interface to the resources defined by the API. (Examples of this approach can be found in the examples-Python github repo, specifically the api_test_service*.py scripts.)

Authorization

Some, but not all, of the endpoints require authorization. This authorization is *not* related to controlled-access data: these endpoints do not operate on or directly return any controlled data. Instead, authorization is related to saving or retrieving cohorts because cohorts are *private* to the user who created the cohort (and anyone the cohort owner has

chosen to share the cohort with). Helper scripts, described below, are provided to access these endpoints from the command line.

Note: Prior to using any endpoints that require authorization, a user must have signed into the web application at least once.

Examples

from Python

Step 1: A python helper-script, isb_auth.py, can be used to start the OAuth flow and store the users credentials in a file named ~/.isb_credentials

```
$ python isb_auth.py
```

This script will open a new tab in your browser and ask you to sign in with your google identity (*eg* your gmail address). The first time, you will also be asked to grant the ISB-CGC application permission to see your email address. Once authenticated, your access and refresh tokens are written to ~/.isb_credentials. You may use the --verbose flag when running this script to see the contents and name of this file.

If you are running this script via ssh (or from Cloud Shell), the --noauth_local_webserver flag will allow you to obtain a verification code through your local browser.

Step 2: Once you have a ~/.isb_credentials file (either locally on your laptop, or on a GCE VM, or in Cloud Shell), you can access any API requiring authentication using another helper-script, isb_curl.py

```
$ ## usage: python isb_curl.py {ENDPOINT_URL}
$ python isb_curl.py https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v2/
↪cohorts
```

from R

The Examples-R (ISBCGCExamples) package contains a number of functions that "wrap" the http endpoints calls, making it easier to access your cohorts and query the database.

Step 1: After starting R, and loading the ISBCGCExamples, you can use the R helper script isb_init to go through the authentication process:

```
> library(ISBCGCExamples)
> token <- isb_init()
Use a local file to cache OAuth access credentials between R sessions?
1: Yes
2: No

Selection: 1
Waiting for authentication in browser...
Press Esc/Ctrl + C to abort
Authentication complete.
```

The isb_init function will open a new tab in your browser and ask you to sign in with your google identity (*eg* your gmail address). The first time, you will also be asked to grant the ISB-CGC application permission to see your email address. Once authenticated, your access and refresh tokens are written to your working directory in a file named .httr-oauth.

Step 2: Using the endpoints

After authentication, any of the example endpoint functions can be used such as:

```
list_cohorts(token)
```

which returns a list of the user's previously created cohorts. Documentation for these functions can be found in the ISB-CGC github repo, Examples-R under 'API Endpoints Interface'.

ISB-CGC API (v3)

The endpoints have been reorganized to support the multiple programs that now have data in the ISB-CGC. These endpoints are now organized into four different sections: TCGA, CCLE, TARGET and common endpoints.

Please Note: For the create.cohort API for all programs require the user to select inbetween the brackets to view the possible filter(s) for cohort being built.

Details for each of these endpoints can be found below:

Universal Endpoints

cohorts().cloud_storage_file_paths()

Takes a cohort id as a required parameter and returns cloud storage paths to files associated with all the samples in that cohort, up to a default limit of 10,000 files. Authentication is required. User must have READER or OWNER permissions on the cohort.

Example:

```
python isb_curl.py https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v3/cohorts/
→{COHORT ID}/cloud_storage_file_paths
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
from oauth2client.client import OAuth2WebServerFlow
from oauth2client import tools
from oauth2client.file import Storage
import httplib2
import os


CLIENT_ID = '907668440978-0ol0griu70qkeb6k3gnn2vipfa5mgl60.apps.googleusercontent.com'
CLIENT_SECRET = 'To_WJH7-1V-TofhNGcEqmEYi'
EMAIL_SCOPE = 'https://www.googleapis.com/auth/userinfo.email'
DEFAULT_STORAGE_FILE = os.path.join(os.path.expanduser('~'), '.isb_credentials')


def get_credentials():
        oauth_flow_args = ['--noauth_local_webserver']
        storage = Storage(DEFAULT_STORAGE_FILE)
        credentials = storage.get()
        if not credentials or credentials.invalid:
                flow = OAuth2WebServerFlow(CLIENT_ID, CLIENT_SECRET, EMAIL_SCOPE)
                flow.auth_uri = flow.auth_uri.rstrip('/') + '?approval_prompt=force'
                credentials = tools.run_flow(flow, storage, tools.argparser.parse_
→args(oauth_flow_args))
```

(continues on next page)

```python
        return credentials


def get_authorized_service():
        api = 'isb_cgc_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site, api,
→version)
        credentials = get_credentials()
        http = credentials.authorize(httplib2.Http())
        if credentials.access_token_expired or credentials.invalid:
                credentials.refresh(http)
        authorized_service = build(api, version, discoveryServiceUrl=discovery_url,
→http=http)
        return authorized_service


service = get_authorized_service()
data = service.cohorts().cloud_storage_file_paths(cohort_id=1).execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v3/cohorts/{cohort_id}/
→cloud_storage_file_paths
```

Parameters

Parameter name	Value	Description
analysis_workflow_type	string	Optional.
cohort_id	string	Required.
data_category	string	Optional.
data_format	string	Optional.
data_type	string	Optional.
experimental_strategy	string	Optional.
genomic_build	string	Optional.
limit	string	Optional.
platform	string	Optional.

Response

If successful, this method returns a response body with the following structure:

```
{
  "cloud_storage_file_paths": [string],
  "count": integer
}
```

Parameter name	Value	Description
cloud_storage_file_paths[]	list	List of Google Cloud Storage paths of files associated with the cohort.
count	integer	Number of Google Cloud Storage paths returned for the cohort.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cohorts().delete()

Deletes a cohort. User must have owner permissions on the cohort.

Example:

```
python isb_curl.py https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v3/cohorts/
↪{COHORT ID} -X DELETE
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
from oauth2client.client import OAuth2WebServerFlow
from oauth2client import tools
from oauth2client.file import Storage
import httplib2
import os

CLIENT_ID = '907668440978-0ol0griu70qkeb6k3gnn2vipfa5mgl60.apps.googleusercontent.com'
CLIENT_SECRET = 'To_WJH7-1V-TofhNGcEqmEYi'
EMAIL_SCOPE = 'https://www.googleapis.com/auth/userinfo.email'
DEFAULT_STORAGE_FILE = os.path.join(os.path.expanduser('~'), '.isb_credentials')


def get_credentials():
        oauth_flow_args = ['--noauth_local_webserver']
        storage = Storage(DEFAULT_STORAGE_FILE)
        credentials = storage.get()
        if not credentials or credentials.invalid:
                flow = OAuth2WebServerFlow(CLIENT_ID, CLIENT_SECRET, EMAIL_SCOPE)
                flow.auth_uri = flow.auth_uri.rstrip('/') + '?approval_prompt=force'
                credentials = tools.run_flow(flow, storage, tools.argparser.parse_
↪args(oauth_flow_args))
        return credentials


def get_authorized_service():
        api = 'isb_cgc_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site, api,
↪version)
        credentials = get_credentials()
        http = credentials.authorize(httplib2.Http())
        if credentials.access_token_expired or credentials.invalid:
                credentials.refresh(http)
        authorized_service = build(api, version, discoveryServiceUrl=discovery_url,
↪http=http)
        return authorized_service

service = get_authorized_service()
data = service.cohorts().delete(cohort_id={YOUR_COHORT_ID}).execute()
```

Request

HTTP request:

```
DELETE https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v3/cohorts/{cohort_id}
```

Parameters

Parameter name	Value	Description
cohort_id	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "message": string
}
```

Parameter name	Value	Description
message	string	Indicates success or failure of cohort deletion.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cohorts().get()

Returns information about a specific cohort the user has READER or OWNER permission on when given a cohort ID. Authentication is required.

Example:

```
python isb_curl.py https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v3/cohorts/
↪{COHORT ID}
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
from oauth2client.client import OAuth2WebServerFlow
from oauth2client import tools
from oauth2client.file import Storage
import httplib2
import os

CLIENT_ID = '907668440978-0ol0griu70qkeb6k3gnn2vipfa5mgl60.apps.googleusercontent.com'
CLIENT_SECRET = 'To_WJH7-1V-TofhNGcEqmEYi'
EMAIL_SCOPE = 'https://www.googleapis.com/auth/userinfo.email'
DEFAULT_STORAGE_FILE = os.path.join(os.path.expanduser('~'), '.isb_credentials')


def get_credentials():
        oauth_flow_args = ['--noauth_local_webserver']
        storage = Storage(DEFAULT_STORAGE_FILE)
        credentials = storage.get()
```

(continues on next page)

```
        if not credentials or credentials.invalid:
                flow = OAuth2WebServerFlow(CLIENT_ID, CLIENT_SECRET, EMAIL_SCOPE)
                flow.auth_uri = flow.auth_uri.rstrip('/') + '?approval_prompt=force'
                credentials = tools.run_flow(flow, storage, tools.argparser.parse_
→args(oauth_flow_args))
        return credentials


def get_authorized_service():
        api = 'isb_cgc_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site, api,
→version)
        credentials = get_credentials()
        http = credentials.authorize(httplib2.Http())
        if credentials.access_token_expired or credentials.invalid:
                credentials.refresh(http)
        authorized_service = build(api, version, discoveryServiceUrl=discovery_url,
→http=http)
        return authorized_service


service = get_authorized_service()
data = service.cohorts().get(cohort_id=1).execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v3/cohorts/{cohort_id}
```

Parameters

Parameter name	Value	Description
cohort_id	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "case_count": integer,
  "cases": [string],
  "comments": string,
  "email": string,
  "filters": [
    {
      "name": string,
      "value": string
    }
  ],
  "id": string,
  "last_date_saved": string,
  "name": string,
  "parent_id": [string],
  "permission": string,
  "sample_count": integer,
```

```
  "samples": [string],
  "source_notes": string,
  "source_type": string
}
```

Parameter name	Value	Description
case_count	integer	Total count of unique case barcodes in the cohort.
cases[]	list	List of case barcodes in the cohort.
comments	string	Comments on the cohort.
email	string	Email of user.
filters[]	list	List of filters applied to create cohort, if any.
filters[].name	string	Names of filtering parameters used to create the cohort.
filters[].value	string	Values of filtering parameters used to create the cohort.
id	string	Cohort id.
last_date_saved	string	Last date the cohort was saved.
name	string	Name of the cohort
parent_id[]	list	List of id's of cohorts that this cohort was derived from, if any.
permission	string	User permissions on cohort: READER or OWNER.
sample_count	integer	Total count of unique sample barcodes in the cohort.
samples[]	list	List of sample barcodes in the cohort.
source_notes	string	Notes on the source of the cohort.
source_type	string	Type of cohort source.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cohorts().list()

Returns information about cohorts a user has either READER or OWNER permission on. Authentication is required.

Example:

```
$ python isb_curl.py https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v3/
↪cohorts
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
from oauth2client.client import OAuth2WebServerFlow
from oauth2client import tools
from oauth2client.file import Storage
import httplib2
import os
```

```python
CLIENT_ID = '907668440978-0ol0griu70qkeb6k3gnn2vipfa5mgl60.apps.googleusercontent.com'
CLIENT_SECRET = 'To_WJH7-1V-TofhNGcEqmEYi'
EMAIL_SCOPE = 'https://www.googleapis.com/auth/userinfo.email'
DEFAULT_STORAGE_FILE = os.path.join(os.path.expanduser('~'), '.isb_credentials')


def get_credentials():
        oauth_flow_args = ['--noauth_local_webserver']
        storage = Storage(DEFAULT_STORAGE_FILE)
        credentials = storage.get()
        if not credentials or credentials.invalid:
                flow = OAuth2WebServerFlow(CLIENT_ID, CLIENT_SECRET, EMAIL_SCOPE)
                flow.auth_uri = flow.auth_uri.rstrip('/') + '?approval_prompt=force'
                credentials = tools.run_flow(flow, storage, tools.argparser.parse_
→args(oauth_flow_args))
        return credentials


def get_authorized_service():
        api = 'isb_cgc_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site, api,
→version)
        credentials = get_credentials()
        http = credentials.authorize(httplib2.Http())
        if credentials.access_token_expired or credentials.invalid:
                credentials.refresh(http)
        authorized_service = build(api, version, discoveryServiceUrl=discovery_url,
→http=http)
        return authorized_service


service = get_authorized_service()
data = service.cohorts().list().execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v3/cohorts
```

Parameters

None

Response

If successful, this method returns a response body with the following structure:

```json
{
  "count": integer,
  "items": [
    {
      "case_count": integer,
      "comments": string,
      "email": string,
      "filters": [
        {
          "name": string,
```

```
        "value": string
      }
    ],
    "id": string,
    "last_date_saved": string,
    "name": string,
    "parent_id": [string],
    "permission": string,
    "sample_count": integer,
    "source_notes": string,
    "source_type": string
  }
 ]
}
```

Parameter name	Value	Description
count	integer	Number of cohorts the user has OWNER or READER permission on.
items[]	list	List of details about each cohort.
items[].case_count	integer	Total count of unique case barcodes in the cohort.
items[].comments	string	Comments on the cohort.
items[].email	string	Email of user.
items[].filters[]	list	List of filters applied to create cohort, if any.
items[].filters[].name	string	Names of filtering parameters used to create the cohort.
items[].filters[].value	string	Values of filtering parameters used to create the cohort.
items[].id	string	Cohort id.
items[].last_date_saved	string	Last date the cohort was saved.
items[].name	string	Name of the cohort
items[].parent_id[]	list	List of id's of cohorts that this cohort was derived from, if any.
items[].permission	string	User permissions on cohort: READER or OWNER.
items[].sample_count	integer	Total count of unique sample barcodes in the cohort.
items[].source_notes	string	Notes on the source of the cohort.
items[].source_type	string	Type of cohort source.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cohorts().file_manifest()

Takes a cohort id as a required parameter and returns a manifest of all files associated with all the samples in that cohort, as well as their metadata, up to a default limit of 50,000 files. Authentication is required. User must have READER or OWNER permissions on the cohort.

Example:

```
python isb_curl.py https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v3/cohorts/
↪{COHORT ID}/file_manifest
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
from oauth2client.client import OAuth2WebServerFlow
from oauth2client import tools
from oauth2client.file import Storage
import httplib2
import os

CLIENT_ID = '907668440978-0ol0griu70qkeb6k3gnn2vipfa5mgl60.apps.googleusercontent.
↪com'
CLIENT_SECRET = 'To_WJH7-1V-TofhNGcEqmEYi'
EMAIL_SCOPE = 'https://www.googleapis.com/auth/userinfo.email'
DEFAULT_STORAGE_FILE = os.path.join(os.path.expanduser('~'), '.isb_credentials')

def get_credentials():
        oauth_flow_args = ['--noauth_local_webserver']
        storage = Storage(DEFAULT_STORAGE_FILE)
        credentials = storage.get()
        if not credentials or credentials.invalid:
                flow = OAuth2WebServerFlow(CLIENT_ID, CLIENT_SECRET, EMAIL_SCOPE)
                flow.auth_uri = flow.auth_uri.rstrip('/') + '?approval_
↪prompt=force'
                credentials = tools.run_flow(flow, storage, tools.argparser.parse_
↪args(oauth_flow_args))
        return credentials


def get_authorized_service():
        api = 'isb_cgc_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site, api,
↪version)
        credentials = get_credentials()
        http = credentials.authorize(httplib2.Http())
        if credentials.access_token_expired or credentials.invalid:
                credentials.refresh(http)
        authorized_service = build(api, version, discoveryServiceUrl=discovery_
↪url, http=http)
        return authorized_service

service = get_authorized_service()
data = service.cohorts().file_manifest(cohort_id=1).execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v3/cohorts/{cohort_id}/
↪file_manifest
```

Parameters

Parameter name	Value	Description
cohort_id	string	Required.
genomic_build	string	Optional, The build for which to obtain file entries, default=hg19
offset	string	Optional, The number of file entries to skip before collecting for retrieval (eg. offset 5 = start at 6), default=0
fetch_count	string	Optional, The number of file entries to retrieve, default=50000

Response

If successful, this method returns a response body with the following structure:

```
{
  "file_list": [
      {
          "program": string,
          "case_barcode": string,
          "case_gdc_uuid": string,
          "file_path": string,
          "file_gdc_uuid": string,
          "disease_code": string,
          "project_short_name": string,
          "experimental_strategy": string,
          "platform": string,
          "data_category": string,
          "data_type": string,
          "data_format": string,
          "access": string
      }
  ],
  "total_file_count": integer,
  "files_retrieved": integer
}
```

Parameter name	Value	Description
file_list[{...}]	list	List of file detail entries of files associated with the cohort.
total_file_count	integer	Total number of file entries found for this cohort.
files_retrieved	integer	Total number of file entries retrieved in this response.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cohorts().file_manifest_filtered()

Takes a cohort id as a required parameter and a filter set as a required data payload, and returns a manifest of all files associated with all the samples in that cohort, as well as their metadata, up to a default limit of 50,000 files. Authentication is required. User must have READER or OWNER permissions on the cohort.

Example:

```
python isb_curl.py --data '{"disease_code": ["LUAD","AML"]}' https://api-dot-isb-cgc.
→appspot.com/_ah/api/isb_cgc_api/v3/cohorts/{COHORT ID}/file_manifest
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
from oauth2client.client import OAuth2WebServerFlow
from oauth2client import tools
from oauth2client.file import Storage
import httplib2
import os

CLIENT_ID = '907668440978-0ol0griu70qkeb6k3gnn2vipfa5mgl60.apps.googleusercontent.com'
CLIENT_SECRET = 'To_WJH7-1V-TofhNGcEqmEYi'
EMAIL_SCOPE = 'https://www.googleapis.com/auth/userinfo.email'
DEFAULT_STORAGE_FILE = os.path.join(os.path.expanduser('~'), '.isb_credentials')


def get_credentials():
        oauth_flow_args = ['--noauth_local_webserver']
        storage = Storage(DEFAULT_STORAGE_FILE)
        credentials = storage.get()
        if not credentials or credentials.invalid:
                flow = OAuth2WebServerFlow(CLIENT_ID, CLIENT_SECRET, EMAIL_SCOPE)
                flow.auth_uri = flow.auth_uri.rstrip('/') + '?approval_prompt=force'
                credentials = tools.run_flow(flow, storage, tools.argparser.parse_
→args(oauth_flow_args))
        return credentials


def get_authorized_service():
        api = 'isb_cgc_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site, api,
→version)
        credentials = get_credentials()
        http = credentials.authorize(httplib2.Http())
        if credentials.access_token_expired or credentials.invalid:
                credentials.refresh(http)
        authorized_service = build(api, version, discoveryServiceUrl=discovery_url,
→http=http)
        return authorized_service

service = get_authorized_service()
data = service.cohorts().file_manifest(cohort_id=1, body={"disease_code": ["BRCA"]}).
→execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_api/v3/cohorts/{cohort_id}/
→file_manifest
```

{

    "disease_code": ["LUSC", "BRCA"

```
        ]
}
```

Parameters

Parameter name	Value	Description
cohort_id	string	Required.
genomic_build	string	Optional, The build for which to obtain file entries, default=hg19
offset	string	Optional, The number of file entries to skip before collecting for retrieval (eg. offset 5 = start at 6), default=0
fetch_count	string	Optional, The number of file entries to retrieve, default=50000
data_category	string	Optional.
data_format	string	Optional.
data_type	string	Optional.
experimental_strategy	string	Optional.
platform	string	Optional.
disease_code	string	Optional.

Response

If successful, this method returns a response body with the following structure:

```
{
  "file_list": [
      {
          "program": string,
          "case_barcode": string,
          "case_gdc_uuid": string,
          "file_path": string,
          "file_gdc_uuid": string,
          "disease_code": string,
          "project_short_name": string,
          "experimental_strategy": string,
          "platform": string,
          "data_category": string,
          "data_type": string,
          "data_format": string,
          "access": string
      }
  ],
  "total_file_count": integer,
  "files_retrieved": integer
}
```

Parameter name	Value	Description
file_list[{...}]	list	List of file detail entries of files associated with the cohort.
total_file_count	integer	Total number of file entries found for this cohort.
files_retrieved	integer	Total number of file entries retrieved in this response.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

TCGA Endpoints

cohorts().preview()

Takes a JSON object of filters in the request body and returns a "preview" of the cohort that would result from passing a similar request to the cohort **save** endpoint. This preview consists of two lists: the lists of case barcodes, and the list of sample barcodes. Authentication is not required.

Example:

```
curl "https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/
↪cohorts/preview?program_short_name=TCGA-UCS&program_short_name=TCGA-CESC&
↪age_at_diagnosis_lte=20"
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_tcga_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
↪http=httplib2.Http())

service = get_unauthorized_service()
body = {'program_short_name': ['TCGA-BRCA', 'TCGA-UCS'], 'age_at_diagnosis_
↪gte': 90}
data = service.cohorts().preview(**body).execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
↪cohorts/preview
```

{

    "Common": {

        "project_short_name": ["TCGA-UCS", "TCGA-CESC"

        ]

    }, "Clinical": {

        "age_at_diagnosis_gte": 20

    }

}

Parameters

Parameter name	Value	Description
age_at_diagnosis	integer	Optional.
age_at_diagnosis_gte	integer	Optional.
age_at_diagnosis_lte	integer	Optional.
age_began_smoking_in_years	integer	Optional.
age_began_smoking_in_years_gte	integer	Optional.
age_began_smoking_in_years_lte	integer	Optional.
anatomic_neoplasm_subdivision	string	Optional.
avg_percent_lymphocyte_infiltration	number	Optional.
avg_percent_lymphocyte_infiltration_gte	number	Optional.
avg_percent_lymphocyte_infiltration_lte	number	Optional.
avg_percent_monocyte_infiltration	number	Optional.
avg_percent_monocyte_infiltration_gte	number	Optional.
avg_percent_monocyte_infiltration_lte	number	Optional.
avg_percent_necrosis	number	Optional.
avg_percent_necrosis_gte	number	Optional.
avg_percent_necrosis_lte	number	Optional.
avg_percent_neutrophil_infiltration	number	Optional.
avg_percent_neutrophil_infiltration_gte	number	Optional.
avg_percent_neutrophil_infiltration_lte	number	Optional.
avg_percent_normal_cells	number	Optional.
avg_percent_normal_cells_gte	number	Optional.
avg_percent_normal_cells_lte	number	Optional.
avg_percent_stromal_cells	number	Optional.
avg_percent_stromal_cells_gte	number	Optional.
avg_percent_stromal_cells_lte	number	Optional.
avg_percent_tumor_cells	number	Optional.
avg_percent_tumor_cells_gte	number	Optional.
avg_percent_tumor_cells_lte	number	Optional.
avg_percent_tumor_nuclei	number	Optional.
avg_percent_tumor_nuclei_gte	number	Optional.
avg_percent_tumor_nuclei_lte	number	Optional.
batch_number	integer	Optional.
batch_number_gte	integer	Optional.
batch_number_lte	integer	Optional.
bcr	string	Optional.
bmi	number	Optional.
bmi_gte	number	Optional.
bmi_lte	number	Optional.
case_barcode	string	Optional.
case_gdc_id	string	Optional.
clinical_M	string	Optional.
clinical_N	string	Optional.
clinical_stage	string	Optional.
clinical_T	string	Optional.
colorectal_cancer	string	Optional.
country	string	Optional.
days_to_birth	integer	Optional.

Table 1 – continued from previous page

Parameter name	Value	Description
days_to_birth_gte	integer	Optional.
days_to_birth_lte	integer	Optional.
days_to_collection	integer	Optional.
days_to_collection_gte	integer	Optional.
days_to_collection_lte	integer	Optional.
days_to_death	integer	Optional.
days_to_death_gte	integer	Optional.
days_to_death_lte	integer	Optional.
days_to_initial_pathologic_diagnosis	integer	Optional.
days_to_initial_pathologic_diagnosis_gte	integer	Optional.
days_to_initial_pathologic_diagnosis_lte	integer	Optional.
days_to_last_followup	integer	Optional.
days_to_last_followup_gte	integer	Optional.
days_to_last_followup_lte	integer	Optional.
days_to_last_known_alive	integer	Optional.
days_to_last_known_alive_gte	integer	Optional.
days_to_last_known_alive_lte	integer	Optional.
days_to_sample_procurement	integer	Optional.
days_to_sample_procurement_gte	integer	Optional.
days_to_sample_procurement_lte	integer	Optional.
days_to_submitted_specimen_dx	integer	Optional.
days_to_submitted_specimen_dx_gte	integer	Optional.
days_to_submitted_specimen_dx_lte	integer	Optional.
disease_code	string	Optional.
endpoint_type	string	Optional.
ethnicity	string	Optional.
gender	string	Optional.
gleason_score_combined	integer	Optional.
gleason_score_combined_gte	integer	Optional.
gleason_score_combined_lte	integer	Optional.
h_pylori_infection	string	Optional.
height	integer	Optional.
height_gte	integer	Optional.
height_lte	integer	Optional.
histological_type	string	Optional.
history_of_colon_polyps	string	Optional.
history_of_neoadjuvant_treatment	string	Optional.
hpv_calls	string	Optional.
hpv_status	string	Optional.
icd_10	string	Optional.
icd_o_3_histology	string	Optional.
icd_o_3_site	string	Optional.
lymphatic_invasion	string	Optional.
lymphnodes_examined	string	Optional.
lymphovascular_invasion_present	string	Optional.
max_percent_lymphocyte_infiltration	number	Optional.
max_percent_lymphocyte_infiltration_gte	number	Optional.
max_percent_lymphocyte_infiltration_lte	number	Optional.
max_percent_monocyte_infiltration	number	Optional.

Table 1 – continued from previous page

Parameter name	Value	Description
max_percent_monocyte_infiltration_gte	number	Optional.
max_percent_monocyte_infiltration_lte	number	Optional.
max_percent_necrosis	number	Optional.
max_percent_necrosis_gte	number	Optional.
max_percent_necrosis_lte	number	Optional.
max_percent_neutrophil_infiltration	number	Optional.
max_percent_neutrophil_infiltration_gte	number	Optional.
max_percent_neutrophil_infiltration_lte	number	Optional.
max_percent_normal_cells	number	Optional.
max_percent_normal_cells_gte	number	Optional.
max_percent_normal_cells_lte	number	Optional.
max_percent_stromal_cells	number	Optional.
max_percent_stromal_cells_gte	number	Optional.
max_percent_stromal_cells_lte	number	Optional.
max_percent_tumor_cells	number	Optional.
max_percent_tumor_cells_gte	number	Optional.
max_percent_tumor_cells_lte	number	Optional.
max_percent_tumor_nuclei	number	Optional.
max_percent_tumor_nuclei_gte	number	Optional.
max_percent_tumor_nuclei_lte	number	Optional.
menopause_status	string	Optional.
min_percent_lymphocyte_infiltration	number	Optional.
min_percent_lymphocyte_infiltration_gte	number	Optional.
min_percent_lymphocyte_infiltration_lte	number	Optional.
min_percent_monocyte_infiltration	number	Optional.
min_percent_monocyte_infiltration_gte	number	Optional.
min_percent_monocyte_infiltration_lte	number	Optional.
min_percent_necrosis	number	Optional.
min_percent_necrosis_gte	number	Optional.
min_percent_necrosis_lte	number	Optional.
min_percent_neutrophil_infiltration	number	Optional.
min_percent_neutrophil_infiltration_gte	number	Optional.
min_percent_neutrophil_infiltration_lte	number	Optional.
min_percent_normal_cells	number	Optional.
min_percent_normal_cells_gte	number	Optional.
min_percent_normal_cells_lte	number	Optional.
min_percent_stromal_cells	number	Optional.
min_percent_stromal_cells_gte	number	Optional.
min_percent_stromal_cells_lte	number	Optional.
min_percent_tumor_cells	number	Optional.
min_percent_tumor_cells_gte	number	Optional.
min_percent_tumor_cells_lte	number	Optional.
min_percent_tumor_nuclei	number	Optional.
min_percent_tumor_nuclei_gte	number	Optional.
min_percent_tumor_nuclei_lte	number	Optional.
mononucleotide_and_dinucleotide_marker_panel_analysis_status	string	Optional.
neoplasm_histologic_grade	string	Optional.
new_tumor_event_after_initial_treatment	string	Optional.
num_portions	integer	Optional.

Table 1 – continued from previous page

Parameter name	Value	Description
num_portions_gte	integer	Optional.
num_portions_lte	integer	Optional.
num_slides	integer	Optional.
num_slides_gte	integer	Optional.
num_slides_lte	integer	Optional.
number_of_lymphnodes_examined	integer	Optional.
number_of_lymphnodes_examined_gte	integer	Optional.
number_of_lymphnodes_examined_lte	integer	Optional.
number_of_lymphnodes_positive_by_he	integer	Optional.
number_of_lymphnodes_positive_by_he_gte	integer	Optional.
number_of_lymphnodes_positive_by_he_lte	integer	Optional.
number_pack_years_smoked	integer	Optional.
number_pack_years_smoked_gte	integer	Optional.
number_pack_years_smoked_lte	integer	Optional.
other_dx	string	Optional.
other_malignancy_anatomic_site	string	Optional.
other_malignancy_histological_type	string	Optional.
other_malignancy_type	string	Optional.
pathologic_M	string	Optional.
pathologic_N	string	Optional.
pathologic_stage	string	Optional.
pathologic_T	string	Optional.
pathology_report_uuid	string	Optional.
person_neoplasm_cancer_status	string	Optional.
pregnancies	string	Optional.
preservation_method	string	Optional.
primary_neoplasm_melanoma_dx	string	Optional.
primary_therapy_outcome_success	string	Optional.
program_name	string	Optional.
project_short_name	string	Optional.
psa_value	number	Optional.
psa_value_gte	number	Optional.
psa_value_lte	number	Optional.
race	string	Optional.
residual_tumor	string	Optional.
sample_barcode	string	Optional.
sample_gdc_id	string	Optional.
sample_type	string	Optional.
stopped_smoking_year	integer	Optional.
stopped_smoking_year_gte	integer	Optional.
stopped_smoking_year_lte	integer	Optional.
summary_file_count	integer	Optional.
summary_file_count_gte	integer	Optional.
summary_file_count_lte	integer	Optional.
tobacco_smoking_history	string	Optional.
tss_code	string	Optional.
tumor_tissue_site	string	Optional.
tumor_type	string	Optional.
venous_invasion	string	Optional.

Table 1 – continued from previous page

Parameter name	Value	Description
vital_status	string	Optional.
weight	integer	Optional.
weight_gte	integer	Optional.
weight_lte	integer	Optional.
year_of_diagnosis	integer	Optional.
year_of_diagnosis_gte	integer	Optional.
year_of_diagnosis_lte	integer	Optional.
year_of_tobacco_smoking_onset	integer	Optional.
year_of_tobacco_smoking_onset_gte	integer	Optional.
year_of_tobacco_smoking_onset_lte	integer	Optional.

Response

If successful, this method returns a response body with the following structure:

```
{
  "case_count": integer,
  "cases": [string],
  "sample_count": integer,
  "samples": [string]
}
```

Parameter name	Value	Description
case_count	integer	Number of cases in the cohort.
cases[]	list	List of cases barcodes in the cohort.
sample_count	integer	Number of samples in the cohort.
samples[]	list	List of sample barcodes in the cohort.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cohorts().create()

Creates and saves a cohort. Takes a JSON object in the request body to use as the cohort's filters. Authentication is required. Returns information about the saved cohort, including the number of cases and the number of samples in that cohort.

Example:

```
python isb_curl.py "https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_
↪api/v3/cohorts/create?name={COHORT NAME}" -H "Content-Type: application/
↪json" -d '{"program_short_name": ["TCGA-UCS", "TCGA-CESC"], "age_at_
↪diagnosis_lte": 60}"
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
from oauth2client.client import OAuth2WebServerFlow
from oauth2client import tools
from oauth2client.file import Storage
import httplib2
import os

CLIENT_ID = '907668440978-0ol0griu70qkeb6k3gnn2vipfa5mgl60.apps.
↪googleusercontent.com'
CLIENT_SECRET = 'To_WJH7-1V-TofhNGcEqmEYi'
EMAIL_SCOPE = 'https://www.googleapis.com/auth/userinfo.email'
DEFAULT_STORAGE_FILE = os.path.join(os.path.expanduser('~'), '.isb_
↪credentials')

def get_credentials():
        oauth_flow_args = ['--noauth_local_webserver']
        storage = Storage(DEFAULT_STORAGE_FILE)
        credentials = storage.get()
        if not credentials or credentials.invalid:
                flow = OAuth2WebServerFlow(CLIENT_ID, CLIENT_SECRET, EMAIL_
↪SCOPE)
                flow.auth_uri = flow.auth_uri.rstrip('/') + '?approval_
↪prompt=force'
                credentials = tools.run_flow(flow, storage, tools.argparser.
↪parse_args(oauth_flow_args))
        return credentials

def get_authorized_service():
        api = 'isb_cgc_tcga_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        credentials = get_credentials()
        http = credentials.authorize(httplib2.Http())
        if credentials.access_token_expired or credentials.invalid:
                credentials.refresh(http)
        authorized_service = build(api, version,
↪discoveryServiceUrl=discovery_url, http=http)
        return authorized_service

service = get_authorized_service()
body = {'program_short_name': ['TCGA-BRCA', 'TCGA-UCS'], 'age_at_diagnosis_
↪gte': 90}
data = service.cohorts().create(name=name, body=body).execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
↪cohorts/create?name=TCGA-test&fields=case_count%2Cfilters%2Cid%2Clast_date_
↪saved%2Cname%2Csample_count
```

{

    "Common": {

        "project_short_name": ["TCGA-UCS", "TCGA-CESC"

```
            ]
      }, "Clinical": {
            "age_at_diagnosis_lte": 60
      }
}
```

Parameters

Parameter name	Value	Description
name	string	Required.

Request body

In the request body, supply a metadata resource with the following properties:

```
{
  "age_at_diagnosis": [integer],
  "age_at_diagnosis_gte": integer,
  "age_at_diagnosis_lte": integer,
  "age_began_smoking_in_years": [integer],
  "age_began_smoking_in_years_gte": integer,
  "age_began_smoking_in_years_lte": integer,
  "anatomic_neoplasm_subdivision": [string],
  "avg_percent_lymphocyte_infiltration": [number],
  "avg_percent_lymphocyte_infiltration_gte": number,
  "avg_percent_lymphocyte_infiltration_lte": number,
  "avg_percent_monocyte_infiltration": [number],
  "avg_percent_monocyte_infiltration_gte": number,
  "avg_percent_monocyte_infiltration_lte": number,
  "avg_percent_necrosis": [number],
  "avg_percent_necrosis_gte": number,
  "avg_percent_necrosis_lte": number,
  "avg_percent_neutrophil_infiltration": [number],
  "avg_percent_neutrophil_infiltration_gte": number,
  "avg_percent_neutrophil_infiltration_lte": number,
  "avg_percent_normal_cells": [number],
  "avg_percent_normal_cells_gte": number,
  "avg_percent_normal_cells_lte": number,
  "avg_percent_stromal_cells": [number],
  "avg_percent_stromal_cells_gte": number,
  "avg_percent_stromal_cells_lte": number,
  "avg_percent_tumor_cells": [number],
  "avg_percent_tumor_cells_gte": number,
  "avg_percent_tumor_cells_lte": number,
  "avg_percent_tumor_nuclei": [number],
  "avg_percent_tumor_nuclei_gte": number,
  "avg_percent_tumor_nuclei_lte": number,
  "batch_number": [integer],
  "batch_number_gte": integer,
  "batch_number_lte": integer,
  "bcr": [string],
  "bmi": [number],
  "bmi_gte": number,
  "bmi_lte": number,
  "case_barcode": [string],
```

```
"case_gdc_id": [string],
"clinical_M": [string],
"clinical_N": [string],
"clinical_stage": [string],
"clinical_T": [string],
"colorectal_cancer": [string],
"country": [string],
"days_to_birth": [integer],
"days_to_birth_gte": integer,
"days_to_birth_lte": integer,
"days_to_collection": [integer],
"days_to_collection_gte": integer,
"days_to_collection_lte": integer,
"days_to_death": [integer],
"days_to_death_gte": integer,
"days_to_death_lte": integer,
"days_to_initial_pathologic_diagnosis": [integer],
"days_to_initial_pathologic_diagnosis_gte": integer,
"days_to_initial_pathologic_diagnosis_lte": integer,
"days_to_last_followup": [integer],
"days_to_last_followup_gte": integer,
"days_to_last_followup_lte": integer,
"days_to_last_known_alive": [integer],
"days_to_last_known_alive_gte": integer,
"days_to_last_known_alive_lte": integer,
"days_to_sample_procurement": [integer],
"days_to_sample_procurement_gte": integer,
"days_to_sample_procurement_lte": integer,
"days_to_submitted_specimen_dx": [integer],
"days_to_submitted_specimen_dx_gte": integer,
"days_to_submitted_specimen_dx_lte": integer,
"disease_code": [string],
"endpoint_type": [string],
"ethnicity": [string],
"gender": [string],
"gleason_score_combined": [integer],
"gleason_score_combined_gte": integer,
"gleason_score_combined_lte": integer,
"h_pylori_infection": [string],
"height": [integer],
"height_gte": integer,
"height_lte": integer,
"histological_type": [string],
"history_of_colon_polyps": [string],
"history_of_neoadjuvant_treatment": [string],
"hpv_calls": [string],
"hpv_status": [string],
"icd_10": [string],
"icd_o_3_histology": [string],
"icd_o_3_site": [string],
"lymphatic_invasion": [string],
"lymphnodes_examined": [string],
"lymphovascular_invasion_present": [string],
"max_percent_lymphocyte_infiltration": [number],
"max_percent_lymphocyte_infiltration_gte": number,
"max_percent_lymphocyte_infiltration_lte": number,
"max_percent_monocyte_infiltration": [number],
```

```
"max_percent_monocyte_infiltration_gte": number,
"max_percent_monocyte_infiltration_lte": number,
"max_percent_necrosis": [number],
"max_percent_necrosis_gte": number,
"max_percent_necrosis_lte": number,
"max_percent_neutrophil_infiltration": [number],
"max_percent_neutrophil_infiltration_gte": number,
"max_percent_neutrophil_infiltration_lte": number,
"max_percent_normal_cells": [number],
"max_percent_normal_cells_gte": number,
"max_percent_normal_cells_lte": number,
"max_percent_stromal_cells": [number],
"max_percent_stromal_cells_gte": number,
"max_percent_stromal_cells_lte": number,
"max_percent_tumor_cells": [number],
"max_percent_tumor_cells_gte": number,
"max_percent_tumor_cells_lte": number,
"max_percent_tumor_nuclei": [number],
"max_percent_tumor_nuclei_gte": number,
"max_percent_tumor_nuclei_lte": number,
"menopause_status": [string],
"min_percent_lymphocyte_infiltration": [number],
"min_percent_lymphocyte_infiltration_gte": number,
"min_percent_lymphocyte_infiltration_lte": number,
"min_percent_monocyte_infiltration": [number],
"min_percent_monocyte_infiltration_gte": number,
"min_percent_monocyte_infiltration_lte": number,
"min_percent_necrosis": [number],
"min_percent_necrosis_gte": number,
"min_percent_necrosis_lte": number,
"min_percent_neutrophil_infiltration": [number],
"min_percent_neutrophil_infiltration_gte": number,
"min_percent_neutrophil_infiltration_lte": number,
"min_percent_normal_cells": [number],
"min_percent_normal_cells_gte": number,
"min_percent_normal_cells_lte": number,
"min_percent_stromal_cells": [number],
"min_percent_stromal_cells_gte": number,
"min_percent_stromal_cells_lte": number,
"min_percent_tumor_cells": [number],
"min_percent_tumor_cells_gte": number,
"min_percent_tumor_cells_lte": number,
"min_percent_tumor_nuclei": [number],
"min_percent_tumor_nuclei_gte": number,
"min_percent_tumor_nuclei_lte": number,
"mononucleotide_and_dinucleotide_marker_panel_analysis_status": [string],
"neoplasm_histologic_grade": [string],
"new_tumor_event_after_initial_treatment": [string],
"num_portions": [integer],
"num_portions_gte": integer,
"num_portions_lte": integer,
"num_slides": [integer],
"num_slides_gte": integer,
"num_slides_lte": integer,
"number_of_lymphnodes_examined": [integer],
"number_of_lymphnodes_examined_gte": integer,
"number_of_lymphnodes_examined_lte": integer,
```

```
    "number_of_lymphnodes_positive_by_he": [integer],
    "number_of_lymphnodes_positive_by_he_gte": integer,
    "number_of_lymphnodes_positive_by_he_lte": integer,
    "number_pack_years_smoked": [integer],
    "number_pack_years_smoked_gte": integer,
    "number_pack_years_smoked_lte": integer,
    "other_dx": [string],
    "other_malignancy_anatomic_site": [string],
    "other_malignancy_histological_type": [string],
    "other_malignancy_type": [string],
    "pathologic_M": [string],
    "pathologic_N": [string],
    "pathologic_stage": [string],
    "pathologic_T": [string],
    "pathology_report_uuid": [string],
    "person_neoplasm_cancer_status": [string],
    "pregnancies": [string],
    "preservation_method": [string],
    "primary_neoplasm_melanoma_dx": [string],
    "primary_therapy_outcome_success": [string],
    "program_name": [string],
    "project_short_name": [string],
    "psa_value": [number],
    "psa_value_gte": number,
    "psa_value_lte": number,
    "race": [string],
    "residual_tumor": [string],
    "sample_barcode": [string],
    "sample_gdc_id": [string],
    "sample_type": [string],
    "stopped_smoking_year": [integer],
    "stopped_smoking_year_gte": integer,
    "stopped_smoking_year_lte": integer,
    "summary_file_count": [integer],
    "summary_file_count_gte": integer,
    "summary_file_count_lte": integer,
    "tobacco_smoking_history": [string],
    "tss_code": [string],
    "tumor_tissue_site": [string],
    "tumor_type": [string],
    "venous_invasion": [string],
    "vital_status": [string],
    "weight": [integer],
    "weight_gte": integer,
    "weight_lte": integer,
    "year_of_diagnosis": [integer],
    "year_of_diagnosis_gte": integer,
    "year_of_diagnosis_lte": integer,
    "year_of_tobacco_smoking_onset": [integer],
    "year_of_tobacco_smoking_onset_gte": integer,
    "year_of_tobacco_smoking_onset_lte": integer
}
```

Parameter name	Value	Description
age_at_diagnosis[]	list	Optional.

Table 2 – continued from previous page

Parameter name	Value	Description
age_at_diagnosis_gte	integer	Optional.
age_at_diagnosis_lte	integer	Optional.
age_began_smoking_in_years[]	list	Optional.
age_began_smoking_in_years_gte	integer	Optional.
age_began_smoking_in_years_lte	integer	Optional.
anatomic_neoplasm_subdivision[]	list	Optional. Possible values include: 'Alveolar Ridge', 'Antrum/Distal', 'Ascending Colon', 'Base of tongue', 'Bilateral', 'Bladder - NOS', 'Body of Pancreas', 'Bronchial', 'Buccal Mucosa', 'Cardia/Proximal', 'Cecum', 'Descending Colon', 'Dome', 'Endometrium', 'Floor of mouth', 'Fundus uteri', 'Fundus/Body', 'Gastroesophageal Junction', 'Hard Palate', 'Head of Pancreas', 'Hepatic Flexure', 'Hypopharynx', 'L-Lower', 'L-Upper', 'Larynx', 'Left', 'Left Lower Inner Quadrant', 'Left Lower Outer Quadrant', 'Left Upper Inner Quadrant', 'Left Upper Outer Quadrant', 'Lip', 'Lower uterine segment/Isthmus uteri', 'Myometrium', 'Neck', 'Oral Cavity', 'Oral Tongue', 'Oropharynx', 'Other (please specify)', 'R-Lower', 'R-Middle', 'R-Upper', 'Rectosigmoid Junction', 'Rectum', 'Right', 'Right Lower Inner Quadrant', 'Right Lower Outer Quadrant', 'Right Upper Inner Quadrant', 'Right Upper Outer Quadrant', 'Sigmoid Colon', 'Splenic Flexure', 'Stomach (NOS)', 'Tail of Pancreas', 'Tonsil', 'Transverse Colon', 'Trigone', 'Unknown - Uterus NOS', 'Wall Anterior', 'Wall Lateral', 'Wall NOS', 'Wall Posterior'.
avg_percent_lymphocyte_infiltration[]	list	Optional.
avg_percent_lymphocyte_infiltration_gte	number	Optional.
avg_percent_lymphocyte_infiltration_lte	number	Optional.
avg_percent_monocyte_infiltration[]	list	Optional.
avg_percent_monocyte_infiltration_gte	number	Optional.
avg_percent_monocyte_infiltration_lte	number	Optional.
avg_percent_necrosis[]	list	Optional.
avg_percent_necrosis_gte	number	Optional.
avg_percent_necrosis_lte	number	Optional.
avg_percent_neutrophil_infiltration[]	list	Optional.
avg_percent_neutrophil_infiltration_gte	number	Optional.
avg_percent_neutrophil_infiltration_lte	number	Optional.
avg_percent_normal_cells[]	list	Optional.
avg_percent_normal_cells_gte	number	Optional.
avg_percent_normal_cells_lte	number	Optional.
avg_percent_stromal_cells[]	list	Optional.
avg_percent_stromal_cells_gte	number	Optional.
avg_percent_stromal_cells_lte	number	Optional.

Continued on next page

Table 2 – continued from previous page

Parameter name	Value	Description
avg_percent_tumor_cells[]	list	Optional.
avg_percent_tumor_cells_gte	number	Optional.
avg_percent_tumor_cells_lte	number	Optional.
avg_percent_tumor_nuclei[]	list	Optional.
avg_percent_tumor_nuclei_gte	number	Optional.
avg_percent_tumor_nuclei_lte	number	Optional.
batch_number[]	list	Optional.
batch_number_gte	integer	Optional.
batch_number_lte	integer	Optional.
bcr[]	list	Optional. Possible values include: 'Nationwide Children's Hospital', 'Washington University'.
bmi[]	list	Optional.
bmi_gte	number	Optional.
bmi_lte	number	Optional.
case_barcode[]	list	Optional.
case_gdc_id[]	list	Optional.
clinical_M[]	list	Optional. Possible values include: 'M0', 'M1', 'M1a', 'M1b', 'M1c', 'MX'.
clinical_N[]	list	Optional. Possible values include: 'N0', 'N1', 'N2', 'N2a', 'N2b', 'N2c', 'N3', 'NX'.
clinical_stage[]	list	Optional. Possible values include: 'Stage I', 'Stage IA', 'Stage IA1', 'Stage IA2', 'Stage IB', 'Stage IB1', 'Stage IB2', 'Stage IC', 'Stage II', 'Stage IIA', 'Stage IIA1', 'Stage IIA2', 'Stage IIB', 'Stage IIC', 'Stage III', 'Stage IIIA', 'Stage IIIB', 'Stage IIIC', 'Stage IIIC1', 'Stage IIIC2', 'Stage IS', 'Stage IV', 'Stage IVA', 'Stage IVB', 'Stage IVC'.
clinical_T[]	list	Optional. Possible values include: 'T1', 'T1a', 'T1b', 'T1c', 'T2', 'T2a', 'T2b', 'T2c', 'T3', 'T3a', 'T3b', 'T4', 'T4a', 'T4b', 'T4c', 'T4d', 'T4e', 'TX'.
colorectal_cancer[]	list	Optional. Possible values include: 'NO', 'YES'.
country[]	list	Optional. Possible values include: 'Afghanistan', 'Algeria', 'American Samoa', 'Australia', 'Brazil', 'Bulgaria', 'Canada', 'Croatia', 'Czech Republic', 'France', 'Georgia', 'Germany', 'Hamburg/Germany', 'Israel', 'Italy', 'Korea', 'Korea South', 'Moldova', 'Netherlands', 'Nigeria', 'Ontario Canada', 'Ontario/Canada', 'Pakistan', 'Poland', 'Puerto Rico', 'Republic of Moldova', 'Romania', 'Russia', 'Sao Paulo', 'Singapore', 'Spain', 'Switzerland', 'Ukraine', 'United Kingdom', 'United States', 'Vietnam', 'Yemen'.
days_to_birth[]	list	Optional.

Continued on next page

Table 2 – continued from previous page

Parameter name	Value	Description
days_to_birth_gte	integer	Optional.
days_to_birth_lte	integer	Optional.
days_to_collection[]	list	Optional.
days_to_collection_gte	integer	Optional.
days_to_collection_lte	integer	Optional.
days_to_death[]	list	Optional.
days_to_death_gte	integer	Optional.
days_to_death_lte	integer	Optional.
days_to_initial_pathologic_diagnosis[]	list	Optional.
days_to_initial_pathologic_diagnosis_gte	integer	Optional.
days_to_initial_pathologic_diagnosis_lte	integer	Optional.
days_to_last_followup[]	list	Optional.
days_to_last_followup_gte	integer	Optional.
days_to_last_followup_lte	integer	Optional.
days_to_last_known_alive[]	list	Optional.
days_to_last_known_alive_gte	integer	Optional.
days_to_last_known_alive_lte	integer	Optional.
days_to_sample_procurement[]	list	Optional.
days_to_sample_procurement_gte	integer	Optional.
days_to_sample_procurement_lte	integer	Optional.
days_to_submitted_specimen_dx[]	list	Optional.
days_to_submitted_specimen_dx_gte	integer	Optional.
days_to_submitted_specimen_dx_lte	integer	Optional.
disease_code[]	list	Optional. Possible values include: 'ACC', 'BLCA', 'BRCA', 'CESC', 'CHOL', 'COAD', 'DLBC', 'ESCA', 'GBM', 'HNSC', 'KICH', 'KIRC', 'KIRP', 'LAML', 'LGG', 'LIHC', 'LUAD', 'LUSC', 'MESO', 'OV', 'PAAD', 'PCPG', 'PRAD', 'READ', 'SARC', 'SKCM', 'STAD', 'TGCT', 'THCA', 'THYM', 'UCEC', 'UCS', 'UVM'.
endpoint_type[]	list	Optional. Possible values include: 'current', 'legacy'.
ethnicity[]	list	Optional. Possible values include: 'HISPANIC OR LATINO', 'NOT HISPANIC OR LATINO'.
gender[]	list	Optional. Possible values include: 'FEMALE', 'MALE'.
gleason_score_combined[]	list	Optional.
gleason_score_combined_gte	integer	Optional.
gleason_score_combined_lte	integer	Optional.
h_pylori_infection[]	list	Optional. Possible values include: 'Current', 'Never', 'No', 'Yes'.
height[]	list	Optional.
height_gte	integer	Optional.
height_lte	integer	Optional.
histological_type[]	list	Optional.
history_of_colon_polyps[]	list	Optional. Possible values include: 'NO', 'YES'.

Continued on next page

Table 2 – continued from previous page

Parameter name	Value	Description
history_of_neoadjuvant_treatment[]	list	Optional. Possible values include: 'No', 'Yes', 'Yes, Pharmaceutical Treatment Prior to Resection', 'Yes, Radiation Prior to Resection'.
hpv_calls[]	list	Optional. Possible values include: 'HPV16', 'HPV16;HPV18', 'HPV16;HPV18;HPV58', 'HPV16;HPV31', 'HPV16;HPV33', 'HPV16;HPV35', 'HPV16;HPV39', 'HPV16;HPV52', 'HPV16;HPV66', 'HPV18', 'HPV18;HPV31', 'HPV31', 'HPV33', 'HPV35', 'HPV39', 'HPV45', 'HPV51', 'HPV52', 'HPV56', 'HPV58', 'HPV59', 'HPV68', 'HPV73'.
hpv_status[]	list	Optional. Possible values include: 'Indeterminate', 'Negative', 'Positive'.
icd_10[]	list	Optional.
icd_o_3_histology[]	list	Optional.
icd_o_3_site[]	list	Optional.
lymphatic_invasion[]	list	Optional. Possible values include: 'NO', 'YES'.
lymphnodes_examined[]	list	Optional. Possible values include: 'NO', 'YES'.
lymphovascular_invasion_present[]	list	Optional. Possible values include: 'NO', 'YES'.
max_percent_lymphocyte_infiltration[]	list	Optional.
max_percent_lymphocyte_infiltration_gte	number	Optional.
max_percent_lymphocyte_infiltration_lte	number	Optional.
max_percent_monocyte_infiltration[]	list	Optional.
max_percent_monocyte_infiltration_gte	number	Optional.
max_percent_monocyte_infiltration_lte	number	Optional.
max_percent_necrosis[]	list	Optional.
max_percent_necrosis_gte	number	Optional.
max_percent_necrosis_lte	number	Optional.
max_percent_neutrophil_infiltration[]	list	Optional.
max_percent_neutrophil_infiltration_gte	number	Optional.
max_percent_neutrophil_infiltration_lte	number	Optional.
max_percent_normal_cells[]	list	Optional.
max_percent_normal_cells_gte	number	Optional.
max_percent_normal_cells_lte	number	Optional.
max_percent_stromal_cells[]	list	Optional.
max_percent_stromal_cells_gte	number	Optional.
max_percent_stromal_cells_lte	number	Optional.
max_percent_tumor_cells[]	list	Optional.
max_percent_tumor_cells_gte	number	Optional.
max_percent_tumor_cells_lte	number	Optional.
max_percent_tumor_nuclei[]	list	Optional.
max_percent_tumor_nuclei_gte	number	Optional.
max_percent_tumor_nuclei_lte	number	Optional.

Table 2 – continued from previous page

Parameter name	Value	Description
menopause_status[]	list	Optional. Possible values include: 'Indeterminate (neither Pre or Postmenopausal)', 'Peri (6-12 months since last menstrual period)', 'Post (prior bilateral ovariectomy OR >12 mo since LMP with no prior hysterectomy)', 'Pre (<6 months since LMP AND no prior bilateral ovariectomy AND not on estrogen replacement)'.
min_percent_lymphocyte_infiltration[]	list	Optional.
min_percent_lymphocyte_infiltration_gte	number	Optional.
min_percent_lymphocyte_infiltration_lte	number	Optional.
min_percent_monocyte_infiltration[]	list	Optional.
min_percent_monocyte_infiltration_gte	number	Optional.
min_percent_monocyte_infiltration_lte	number	Optional.
min_percent_necrosis[]	list	Optional.
min_percent_necrosis_gte	number	Optional.
min_percent_necrosis_lte	number	Optional.
min_percent_neutrophil_infiltration[]	list	Optional.
min_percent_neutrophil_infiltration_gte	number	Optional.
min_percent_neutrophil_infiltration_lte	number	Optional.
min_percent_normal_cells[]	list	Optional.
min_percent_normal_cells_gte	number	Optional.
min_percent_normal_cells_lte	number	Optional.
min_percent_stromal_cells[]	list	Optional.
min_percent_stromal_cells_gte	number	Optional.
min_percent_stromal_cells_lte	number	Optional.
min_percent_tumor_cells[]	list	Optional.
min_percent_tumor_cells_gte	number	Optional.
min_percent_tumor_cells_lte	number	Optional.
min_percent_tumor_nuclei[]	list	Optional.
min_percent_tumor_nuclei_gte	number	Optional.
min_percent_tumor_nuclei_lte	number	Optional.
mononucleotide_and_dinucleotide_marker_panel_analysis_status[]	list	Optional. Possible values include: 'Indeterminate', 'MSI-H', 'MSI-L', 'MSS', 'Not Tested'.
neoplasm_histologic_grade[]	list	Optional. Possible values include: 'G1', 'G2', 'G3', 'G4', 'GB', 'GX', 'High Grade', 'Low Grade'.
new_tumor_event_after_initial_treatment[]	list	Optional. Possible values include: 'NO', 'YES'.
num_portions[]	list	Optional.
num_portions_gte	integer	Optional.
num_portions_lte	integer	Optional.
num_slides[]	list	Optional.
num_slides_gte	integer	Optional.
num_slides_lte	integer	Optional.
number_of_lymphnodes_examined[]	list	Optional.
number_of_lymphnodes_examined_gte	integer	Optional.
number_of_lymphnodes_examined_lte	integer	Optional.
number_of_lymphnodes_positive_by_he[]	list	Optional.

Continued on next page

Table 2 – continued from previous page

Parameter name	Value	Description
number_of_lymphnodes_positive_by_he_gte	integer	Optional.
number_of_lymphnodes_positive_by_he_lte	integer	Optional.
number_pack_years_smoked[]	list	Optional.
number_pack_years_smoked_gte	integer	Optional.
number_pack_years_smoked_lte	integer	Optional.
other_dx[]	list	Optional. Possible values include: 'Both History of Synchronous/ Bilateral and Prior Malignancy', 'No', 'Yes, History of Prior Malignancy', 'Yes, History of Synchronous/Bilateral Malignancy'.
other_malignancy_anatomic_site[]	list	Optional.

Continued on next page

Table 2 – continued from previous page

Parameter name	Value	Description
other_malignancy_histological_type[]	list	Optional. Possible values include: 'Adenocarcinoma, Not Otherwise Specified', 'Adenocarcinoma, Not Otherwise Specified, Adenocarcinoma, Not Otherwise Specified', 'Adenocarcinoma, Not Otherwise Specified, Colon Adenocarcinoma', 'Adenocarcinoma, Not Otherwise Specified, Kidney Clear Cell Renal Carcinoma', 'Adenocarcinoma, Not Otherwise Specified, Lung Acinar Adenocarcinoma', 'Adenocarcinoma, Not Otherwise Specified, Other, specify', 'Adenocarcinoma, Not Otherwise Specified, Other, specify, Other, specify', 'Adenocarcinoma, Not Otherwise Specified, Squamous Cell Carcinoma, Not Otherwise Specified', 'Adenosquamous', 'Astrocytoma', 'Basaloid Squamous Cell', 'Basaloid Squamous Cell, Adenocarcinoma, Not Otherwise Specified', 'Clear Cell Adenocarcinoma', 'Clear Cell Squamous Cell', 'Colon Adenocarcinoma', 'Colon Adenocarcinoma, Colon Adenocarcinoma', 'Colon Mucinous Adenocarcinoma', 'Endometrioid endometrial adenocarcinoma (Grade 1 or 2)', 'Endometrioid endometrial adenocarcinoma (Grade 3)', 'Head & Neck Squamous Cell Carcinoma', 'Hepatocellular Carcinoma', 'Kidney Clear Cell Renal Carcinoma', 'Kidney Clear Cell Renal Carcinoma, Kidney Clear Cell Renal Carcinoma', 'Kidney Clear Cell Renal Carcinoma, Kidney Clear Cell Renal Carcinoma, Other, specify', 'Kidney Clear Cell Renal Carcinoma, Kidney Papillary Renal Cell Carcinoma', 'Kidney Clear Cell Renal Carcinoma, Other, specify', 'Kidney Papillary Renal Cell Carcinoma', 'Kidney Papillary Renal Cell Carcinoma, Kidney Papillary Renal Cell Carcinoma', 'Kidney Papillary Renal Cell Carcinoma, Kidney Papillary Renal Cell Carcinoma, Adenocarcinoma, Not Otherwise Specified', 'Lung Adenocarcinoma Mixed Subtype', 'Lung Adenocarcinoma- Not Otherwise Specified (NOS)', 'Lung Adenocarcinoma- Not Otherwise Specified (NOS), Adenocarcinoma, Not Otherwise Specified', 'Lung Bronchioloalveolar Carcinoma Nonmucinous', 'Lung Clear Cell Squamous Cell Carcinoma', 'Lung Clear Cell Squamous Cell Carcinoma, Other, specify', 'Lung Papillary Adenocarcinoma', 'Lung Small Cell Squamous Cell Carcinoma', 'Other, specify', 'Other, specify, Adenocarcinoma, Not Otherwise Specified', 'Other, specify, Adenocarcinoma, Not Otherwise Specified, Other, specify', 'Other, specify, Basaloid Squamous Cell', 'Other, specify, Clear Cell Adenocarcinoma', 'Other, specify, Kidney

Table 2 – continued from previous page

Parameter name	Value	Description
other_malignancy_type[]	list	Optional. Possible values include: 'Prior Malignancy', 'Prior Malignancy, Prior Malignancy', 'Prior Malignancy, Prior Malignancy, Prior Malignancy', 'Prior Malignancy, Prior Malignancy, Prior Malignancy, Synchronous Malignancy', 'Prior Malignancy, Prior Malignancy, Synchronous Malignancy', 'Prior Malignancy, Synchronous Malignancy', 'Prior Malignancy, Synchronous Malignancy, Prior Malignancy', 'Synchronous Malignancy', 'Synchronous Malignancy, Prior Malignancy', 'Synchronous Malignancy, Prior Malignancy, Prior Malignancy, Prior Malignancy', 'Synchronous Malignancy, Prior Malignancy, Synchronous Malignancy', 'Synchronous Malignancy, Synchronous Malignancy', 'Synchronous Malignancy, Synchronous Malignancy, Prior Malignancy'.
pathologic_M[]	list	Optional. Possible values include: 'cM0 (i+)', 'M0', 'M1', 'M1a', 'M1b', 'M1c', 'MX'.
pathologic_N[]	list	Optional. Possible values include: 'N0', 'N0 (i+)', 'N0 (i-)', 'N0 (mol+)', 'N1', 'N1a', 'N1b', 'N1c', 'N1mi', 'N2', 'N2a', 'N2b', 'N2c', 'N3', 'N3a', 'N3b', 'N3c', 'NX'.
pathologic_stage[]	list	Optional. Possible values include: 'I/II NOS', 'IS', 'Stage 0', 'Stage I', 'Stage IA', 'Stage IB', 'Stage II', 'Stage IIA', 'Stage IIB', 'Stage IIC', 'Stage III', 'Stage IIIA', 'Stage IIIB', 'Stage IIIC', 'Stage IV', 'Stage IVA', 'Stage IVB', 'Stage IVC', 'Stage X'.
pathologic_T[]	list	Optional. Possible values include: 'T0', 'T1', 'T1a', 'T1a1', 'T1b', 'T1b1', 'T1b2', 'T1c', 'T2', 'T2a', 'T2a1', 'T2a2', 'T2b', 'T2c', 'T3', 'T3a', 'T3b', 'T3c', 'T4', 'T4a', 'T4b', 'T4c', 'T4d', 'T4e', 'Tis', 'TX'.
pathology_report_uuid[]	list	Optional.
person_neoplasm_cancer_status[]	list	Optional. Possible values include: 'TUMOR FREE', 'WITH TUMOR'.
pregnancies[]	list	Optional. Possible values include: '0', '1', '2', '3', '4+'.
preservation_method[]	list	Optional. Possible values include: 'FFPE', 'frozen'.
primary_neoplasm_melanoma_dx[]	list	Optional. Possible values include: 'NO', 'YES'.

Continued on next page

Table 2 – continued from previous page

Parameter name	Value	Description
primary_therapy_outcome_success[]	list	Optional. Possible values include: 'Complete Remission/Response', 'No Measureable Tumor or Tumor Markers', 'Normalization of Tumor Markers, but Residual Tumor Mass', 'Partial Remission/Response', 'Persistent Disease', 'Progressive Disease', 'Stable Disease'.
program_name[]	list	Optional. Possible values include: 'TCGA'.
project_short_name[]	list	Optional. Possible values include: 'TCGA-ACC', 'TCGA-BLCA', 'TCGA-BRCA', 'TCGA-CESC', 'TCGA-CHOL', 'TCGA-COAD', 'TCGA-DLBC', 'TCGA-ESCA', 'TCGA-GBM', 'TCGA-HNSC', 'TCGA-KICH', 'TCGA-KIRC', 'TCGA-KIRP', 'TCGA-LAML', 'TCGA-LGG', 'TCGA-LIHC', 'TCGA-LUAD', 'TCGA-LUSC', 'TCGA-MESO', 'TCGA-OV', 'TCGA-PAAD', 'TCGA-PCPG', 'TCGA-PRAD', 'TCGA-READ', 'TCGA-SARC', 'TCGA-SKCM', 'TCGA-STAD', 'TCGA-TGCT', 'TCGA-THCA', 'TCGA-THYM', 'TCGA-UCEC', 'TCGA-UCS', 'TCGA-UVM'.
psa_value[]	list	Optional.
psa_value_gte	number	Optional.
psa_value_lte	number	Optional.
race[]	list	Optional. Possible values include: 'AMERICAN INDIAN OR ALASKA NATIVE', 'ASIAN', 'BLACK OR AFRICAN AMERICAN', 'NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER', 'WHITE'.
residual_tumor[]	list	Optional. Possible values include: 'R0', 'R1', 'R2', 'RX'.
sample_barcode[]	list	Optional.
sample_gdc_id[]	list	Optional.
sample_type[]	list	Optional. Possible values include: '01', '02', '03', '05', '06', '07', '10', '11', '12', '14'.
stopped_smoking_year[]	list	Optional.
stopped_smoking_year_gte	integer	Optional.
stopped_smoking_year_lte	integer	Optional.
summary_file_count[]	list	Optional.
summary_file_count_gte	integer	Optional.
summary_file_count_lte	integer	Optional.
tobacco_smoking_history[]	list	Optional. Possible values include: '1', '2', '3', '4', '5'.
tss_code[]	list	Optional.
tumor_tissue_site[]	list	Optional.
tumor_type[]	list	Optional. Possible values include: 'Primary', 'Type 1', 'Type 2'.
venous_invasion[]	list	Optional. Possible values include: 'NO', 'YES'.

Continued on next page

Table 2 – continued from previous page

Parameter name	Value	Description
vital_status[]	list	Optional. Possible values include: 'Alive', 'Dead'.
weight[]	list	Optional.
weight_gte	integer	Optional.
weight_lte	integer	Optional.
year_of_diagnosis[]	list	Optional.
year_of_diagnosis_gte	integer	Optional.
year_of_diagnosis_lte	integer	Optional.
year_of_tobacco_smoking_onset[]	list	Optional.
year_of_tobacco_smoking_onset_gte	integer	Optional.
year_of_tobacco_smoking_onset_lte	integer	Optional.

Response

If successful, this method returns a response body with the following structure:

```
{
  "case_count": integer,
  "filters": [
    {
      "name": string,
      "value": string
    }
  ],
  "id": string,
  "last_date_saved": string,
  "name": string,
  "sample_count": integer
}
```

Parameter name	Value	Description
case_count	integer	Number of unique case barcodes in the cohort.
filters[]	list	List of filters applied to create cohort, if any.
filters[].name	string	Names of filtering parameters used to create the cohort.
filters[].value	string	Values of filtering parameters used to create the cohort.
id	string	Cohort id.
last_date_saved	string	Last date the cohort was saved.
name	string	Name of cohort.
sample_count	integer	Number of unique sample barcodes in the cohort.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cases().get()

Returns information about a specific case, including a list of samples and aliquots derived from this case. Takes a case barcode (of length 12, *eg* TCGA-B9-7268) as a required parameter. User does not need to be

authenticated.

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
↪cases/TCGA-ZH-A8Y6
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_tcga_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
↪http=httplib2.Http())

service = get_unauthorized_service()
data = service.cases().get(case_barcode='TCGA-W5-AA2R').execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
↪cases/{case_barcode}
```

Parameters

Parameter name	Value	Description
case_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "aliquots": [string],
  "clinical_data": {
    "age_at_diagnosis": integer,
    "age_began_smoking_in_years": integer,
    "anatomic_neoplasm_subdivision": string,
    "avg_percent_lymphocyte_infiltration": number,
    "avg_percent_monocyte_infiltration": number,
    "avg_percent_necrosis": number,
    "avg_percent_neutrophil_infiltration": number,
    "avg_percent_normal_cells": number,
    "avg_percent_stromal_cells": number,
    "avg_percent_tumor_cells": number,
    "avg_percent_tumor_nuclei": number,
    "batch_number": integer,
```

(continues on next page)

```
    "bcr": string,
    "bmi": number,
    "case_barcode": string,
    "case_gdc_id": string,
    "clinical_M": string,
    "clinical_N": string,
    "clinical_stage": string,
    "clinical_T": string,
    "colorectal_cancer": string,
    "country": string,
    "days_to_birth": integer,
    "days_to_collection": integer,
    "days_to_death": integer,
    "days_to_initial_pathologic_diagnosis": integer,
    "days_to_last_followup": integer,
    "days_to_last_known_alive": integer,
    "days_to_sample_procurement": integer,
    "days_to_submitted_specimen_dx": integer,
    "disease_code": string,
    "endpoint_type": string,
    "ethnicity": string,
    "gender": string,
    "gleason_score_combined": integer,
    "h_pylori_infection": string,
    "height": integer,
    "histological_type": string,
    "history_of_colon_polyps": string,
    "history_of_neoadjuvant_treatment": string,
    "hpv_calls": string,
    "hpv_status": string,
    "icd_10": string,
    "icd_o_3_histology": string,
    "icd_o_3_site": string,
    "lymphatic_invasion": string,
    "lymphnodes_examined": string,
    "lymphovascular_invasion_present": string,
    "max_percent_lymphocyte_infiltration": number,
    "max_percent_monocyte_infiltration": number,
    "max_percent_necrosis": number,
    "max_percent_neutrophil_infiltration": number,
    "max_percent_normal_cells": number,
    "max_percent_stromal_cells": number,
    "max_percent_tumor_cells": number,
    "max_percent_tumor_nuclei": number,
    "menopause_status": string,
    "min_percent_lymphocyte_infiltration": number,
    "min_percent_monocyte_infiltration": number,
    "min_percent_necrosis": number,
    "min_percent_neutrophil_infiltration": number,
    "min_percent_normal_cells": number,
    "min_percent_stromal_cells": number,
    "min_percent_tumor_cells": number,
    "min_percent_tumor_nuclei": number,
    "mononucleotide_and_dinucleotide_marker_panel_analysis_status": string,
    "neoplasm_histologic_grade": string,
    "new_tumor_event_after_initial_treatment": string,
    "num_portions": integer,
```

```
    "num_slides": integer,
    "number_of_lymphnodes_examined": integer,
    "number_of_lymphnodes_positive_by_he": integer,
    "number_pack_years_smoked": integer,
    "other_dx": string,
    "other_malignancy_anatomic_site": string,
    "other_malignancy_histological_type": string,
    "other_malignancy_type": string,
    "pathologic_M": string,
    "pathologic_N": string,
    "pathologic_stage": string,
    "pathologic_T": string,
    "pathology_report_uuid": string,
    "person_neoplasm_cancer_status": string,
    "pregnancies": string,
    "preservation_method": string,
    "primary_neoplasm_melanoma_dx": string,
    "primary_therapy_outcome_success": string,
    "program_name": string,
    "project_short_name": string,
    "psa_value": number,
    "race": string,
    "residual_tumor": string,
    "sample_barcode": string,
    "sample_gdc_id": string,
    "sample_type": string,
    "stopped_smoking_year": integer,
    "summary_file_count": integer,
    "tobacco_smoking_history": string,
    "tss_code": string,
    "tumor_tissue_site": string,
    "tumor_type": string,
    "venous_invasion": string,
    "vital_status": string,
    "weight": integer,
    "year_of_diagnosis": integer,
    "year_of_tobacco_smoking_onset": integer
  },
  "samples": [string]
}
```

Parameter name	Value	Description
aliquots[]	list	List of barcodes of aliquots taken from this patient.
clinical_data	nested object	The clinical data about the patient.
clinical_data.age_at_diagnosis	integer	Age at which a condition or disease was first diagnosed in years.
clinical_data.age_began_smoking_in_years	integer	Age began smoking cigarettes expressed in number of years since birth.
clinical_data.anatomic_neoplasm_subdivision	string	Text term to describe the spatial location, subdivisions and/or anatomic site name of a tumor.

Continued on next page

Table 3 – continued from previous page

Parameter name	Value	Description
clinical_data.avg_percent_lymphocyte_infiltration	number	Average in the series of numeric values to represent the percentage of lymphocyte infiltration in a malignant tumor sample or specimen.
clinical_data.avg_percent_monocyte_infiltration	number	Average in the series of numeric values to represent the percentage of monocyte infiltration in a malignant tumor sample or specimen.
clinical_data.avg_percent_necrosis	number	Average in the series of numeric values to represent the percentage of cell death in a malignant tumor sample or specimen.
clinical_data.avg_percent_neutrophil_infiltration	number	Average in the series of numeric values to represent the percentage of neutrophil infiltration in a malignant tumor sample or specimen.
clinical_data.avg_percent_normal_cells	number	Average in the series of numeric values to represent the percentage of normal cells in a malignant tumor sample or specimen.
clinical_data.avg_percent_stromal_cells	number	Average in the series of numeric values to represent the percentage of stromal cells in a malignant tumor sample or specimen.
clinical_data.avg_percent_tumor_cells	number	Average in the series of numeric values to represent the percentage of tumor cells in a malignant tumor sample or specimen.
clinical_data.avg_percent_tumor_nuclei	number	Average in the series of numeric values to represent the percentage of tumor nuclei in a malignant tumor sample or specimen.
clinical_data.batch_number	integer	Groups samples by the batch they were processed in.
clinical_data.bcr	string	A TCGA center where samples are carefully catalogued, processed, quality-checked and stored along with participant clinical information.
clinical_data.bmi	number	Body Mass Index
clinical_data.case_barcode	string	Case barcode.
clinical_data.case_gdc_id	string	The GDC assigned id for the case
clinical_data.clinical_M	string	Extent of the distant metastasis for the cancer based on evidence obtained from clinical assessment parameters determined prior to treatment.
clinical_data.clinical_N	string	Extent of the regional lymph node involvement for the cancer based on evidence obtained from clinical assessment parameters determined prior to treatment.
clinical_data.clinical_stage	string	Stage group determined from clinical information on the tumor (T), regional node (N) and metastases (M) and by grouping cases with similar prognosis.

Continued on next page

Table 3 – continued from previous page

Parameter name	Value	Description
clinical_data.clinical_T	string	Extent of the primary cancer based on evidence obtained from clinical assessment parameters determined prior to treatment.
clinical_data.colorectal_cancer	string	Text term to signify whether a patient has been diagnosed with colorectal cancer.
clinical_data.country	string	Text to identify the name of the state, province, or country in which the sample was procured.
clinical_data.days_to_birth	integer	Time interval from a person's date of birth to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_collection	integer	The number of days between diagnosis and tissue collection.
clinical_data.days_to_death	integer	Time interval from a person's date of death to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_initial_pathologic_diagnosis	integer	Numeric value to represent the day of an individual's initial pathologic diagnosis of cancer.
clinical_data.days_to_last_followup	integer	Time interval from the date of last followup to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_last_known_alive	integer	The number of days between diagnosis and when the individual was last known to be alive.
clinical_data.days_to_sample_procurement	integer	Indicates the days to sample procurement for the submitted sample in relation to the date of initial diagnosis
clinical_data.days_to_submitted_specimen_dx	integer	Time interval from the date of diagnosis of the submitted sample to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.disease_code	string	Text term referring to the cancer type
clinical_data.endpoint_type	string	Which type of GDC Case API was used, either legacy or current
clinical_data.ethnicity	string	The text for reporting information about ethnicity based on the Office of Management and Budget (OMB) categories.
clinical_data.gender	string	Text designations that identify gender.
clinical_data.gleason_score_combined	integer	A numeric value obtained by adding the primary and secondary patterns (grades).
clinical_data.h_pylori_infection	string	Text term to indicate the state of the diagnosis of an individual with Helicobacter pylori infection.
clinical_data.height	integer	The height of the patient in centimeters.
clinical_data.histological_type	string	Text term for the structural pattern of cancer cells used to define a microscopic diagnosis.

Continued on next page

Table 3 – continued from previous page

Parameter name	Value	Description
clinical_data.history_of_colon_polyps	string	Yes/No indicator to describe if the subject had a previous history of colon polyps as noted in the history/physical or previous endoscopic report(s).
clinical_data.history_of_neoadjuvant_treatment	string	Text term to describe the patient's history of neoadjuvant treatment and the kind of treatment given prior to resection of the tumor.
clinical_data.hpv_calls	string	Results of HPV tests.
clinical_data.hpv_status	string	Current HPV status.
clinical_data.icd_10	string	The tenth version of the International Classification of Disease (ICD).
clinical_data.icd_o_3_histology	string	The third edition of the International Classification of Diseases for Oncology.
clinical_data.icd_o_3_site	string	The third edition of the International Classification of Diseases for Oncology.
clinical_data.lymphatic_invasion	string	A yes/no indicator to ask if malignant cells are present in small or thin-walled vessels suggesting lymphatic involvement.
clinical_data.lymphnodes_examined	string	A yes/no/unknown indicator whether a lymph node assessment was performed at the primary presentation of disease.
clinical_data.lymphovascular_invasion_present	string	A yes/no indicator to ask if large vessel (vascular) invasion or small, thin-walled (lymphatic) invasion was detected in a tumor specimen.
clinical_data.max_percent_lymphocyte_infiltration	number	Maximum in the series of numeric values to represent the percentage of lymphocyte infiltration in a malignant tumor sample or specimen.
clinical_data.max_percent_monocyte_infiltration	number	Maximum in the series of numeric values to represent the percentage of monocyte infiltration in a malignant tumor sample or specimen.
clinical_data.max_percent_necrosis	number	Maximum in the series of numeric values to represent the percentage of cell death in a malignant tumor sample or specimen.
clinical_data.max_percent_neutrophil_infiltration	number	Maximum in the series of numeric values to represent the percentage of neutrophil infiltration in a malignant tumor sample or specimen.
clinical_data.max_percent_normal_cells	number	Maximum in the series of numeric values to represent the percentage of normal cells in a malignant tumor sample or specimen.
clinical_data.max_percent_stromal_cells	number	Maximum in the series of numeric values to represent the percentage of stromal cells in a malignant tumor sample or specimen.
clinical_data.max_percent_tumor_cells	number	Maximum in the series of numeric values to represent the percentage of tumor cells in a malignant tumor sample or specimen.

Table 3 – continued from previous page

Parameter name	Value	Description
clinical_data.max_percent_tumor_nuclei	number	Maximum in the series of numeric values to represent the percentage of tumor nuclei in a malignant tumor sample or specimen.
clinical_data.menopause_status	string	Text term to signify the status of a woman's menopause, the permanent cessation of menses, usually defined by 6 to 12 months of amenorrhea.
clinical_data.min_percent_lymphocyte_infiltration	number	Minimum in the series of numeric values to represent the percentage of lymphcyte infiltration in a malignant tumor sample or specimen.
clinical_data.min_percent_monocyte_infiltration	number	Minimum in the series of numeric values to represent the percentage of monocyte infiltration in a malignant tumor sample or specimen.
clinical_data.min_percent_necrosis	number	Minimum in the series of numeric values to represent the percentage of cell death in a malignant tumor sample or specimen.
clinical_data.min_percent_neutrophil_infiltration	number	Minimum in the series of numeric values to represent the percentage of neutrophil infiltration in a malignant tumor sample or specimen.
clinical_data.min_percent_normal_cells	number	Minimum in the series of numeric values to represent the percentage of normal cells in a malignant tumor sample or specimen.
clinical_data.min_percent_stromal_cells	number	Minimum in the series of numeric values to represent the percentage of stromal cells in a malignant tumor sample or specimen.
clinical_data.min_percent_tumor_cells	number	Minimum in the series of numeric values to represent the percentage of tumor cells in a malignant tumor sample or specimen.
clinical_data.min_percent_tumor_nuclei	number	Minimum in the series of numeric values to represent the percentage of tumor nuclei in a malignant tumor sample or specimen.
clinical_data.mononucleotide_and_dinucleotide_marker_panel_analysis_status	string	Text result of microsatellite instability (MSI) testing at using a mononucleotide and dinucleotide microsatellite panel.
clinical_data.neoplasm_histologic_grade	string	Numeric value to express the degree of abnormality of cancer cells, a measure of differentiation and aggressiveness.
clinical_data.new_tumor_event_after_initial_treatment	string	Yes/No/Unknown indicator to identify whether a patient has had a new tumor event after initial treatment.
clinical_data.num_portions	integer	The number of portions obtained from the sample
clinical_data.num_slides	integer	The number of slides derived from the sample
clinical_data.number_of_lymphnodes_examined	integer	The total number of lymph nodes removed and pathologically assessed for disease.

Continued on next page

Table 3 – continued from previous page

Parameter name	Value	Description
clinical_data.number_of_lymphnodes_positive_by_he	integer	Numeric value to signify the count of positive lymph nodes identified through hematoxylin and eosin (H&E) staining light microscopy.
clinical_data.number_pack_years_smoked	integer	Numeric computed value to represent lifetime tobacco exposure defined as number of cigarettes smoked per day x number of years smoked divided by 20.
clinical_data.other_dx	string	Text term to describe the patient's history of cancer diagnosis and the spatial location of any previous cancer occurrence.
clinical_data.other_malignancy_anatomic_site	string	Text term describe the anatomic site of the prior or synchronous malignancy.
clinical_data.other_malignancy_histological_type	string	Text term describe the histology and/or subtype of the prior or synchronous malignancy.
clinical_data.other_malignancy_type	string	The type, relative occurance to the current malignancy
clinical_data.pathologic_M	string	Code to represent the defined absence or presence of distant spread or metastases (M) to locations via vascular channels or lymphatics beyond the regional lymph nodes, using criteria established by the American Joint Committee on Cancer (AJCC).
clinical_data.pathologic_N	string	The codes that represent the stage of cancer based on the nodes present (N stage) according to criteria based on multiple editions of the AJCC's Cancer Staging Manual.
clinical_data.pathologic_stage	string	The extent of a cancer, especially whether the disease has spread from the original site to other parts of the body based on AJCC staging criteria.
clinical_data.pathologic_T	string	Code of pathological T (primary tumor) to define the size or contiguous extension of the primary tumor (T), using staging criteria from the American Joint Committee on Cancer (AJCC).
clinical_data.pathology_report_uuid	string	The UUID of th epathology report
clinical_data.person_neoplasm_cancer_status	string	The state or condition of an individual's neoplasm at a particular point in time.
clinical_data.pregnancies	string	Value to describe the number of full-term pregnancies that a woman has experienced.
clinical_data.preservation_method	string	The method used to preserve the sample after it has been removed from a participant.
clinical_data.primary_neoplasm_melanoma_dx	string	Text indicator to signify whether a person had a primary diagnosis of melanoma.
clinical_data.primary_therapy_outcome_success	string	Measure of success.
clinical_data.program_name	string	Project name, e.g. 'TCGA'.
clinical_data.project_short_name	string	Tumor type abbreviation, e.g. 'BRCA'.

Continued on next page

Table 3 – continued from previous page

Parameter name	Value	Description
clinical_data.psa_value	number	The lab value that represents the results of the most recent (post-operative) prostatic-specific antigen (PSA) in the blood.
clinical_data.race	string	The text for reporting information about race based on the Office of Management and Budget (OMB) categories.
clinical_data.residual_tumor	string	Text terms to describe the status of a tissue margin following surgical resection.
clinical_data.sample_barcode	string	The barcode assigned by TCGA to a sample from a Participant.
clinical_data.sample_gdc_id	string	The GDC assigned id for the sample
clinical_data.sample_type	string	The type of the sample tumor or normal tissue cell or blood sample provided by a participant.
clinical_data.stopped_smoking_year	integer	The year in which the participant quit smoking.
clinical_data.summary_file_count	integer	The count of files associated with the sample
clinical_data.tobacco_smoking_history	string	Category describing current smoking status and smoking history as self-reported by a patient.
clinical_data.tss_code	string	A TSS ID is an alphanumeric code that uniquely identifies a TSS and its associated study
clinical_data.tumor_tissue_site	string	Text term that describes the anatomic site of the tumor or disease.
clinical_data.tumor_type	string	Text term to identify the morphologic sub-type of papillary renal cell carcinoma.
clinical_data.venous_invasion	string	The result of an assessment using the Weiss histopathologic criteria.
clinical_data.vital_status	string	The survival state of the person registered on the protocol.
clinical_data.weight	integer	The weight of the patient measured in kilograms.
clinical_data.year_of_diagnosis	integer	Numeric value to represent the year of an individual's initial pathologic diagnosis of cancer.
clinical_data.year_of_tobacco_smoking_onset	integer	The year in which the participant began smoking.
samples[]	list	List of barcodes of samples taken from this patient.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cases().get_list()

Given a list of case barcodes (of length 11, *eg* TCGA-B9-7268), this endpoint returns all available "biospecimen" information about the cases, including a list of samples and aliquots derived from them. Takes a list of case barcodes (of length 12, *eg* TCGA-B9-7268) as a required data payload. User does not need to be authenticated.

Example:

```
curl --data '{"case_barcodes": ["TCGA-ZH-A8Y6","TCGA-B9-7268"]}' https://api-
→dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/cases
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_tcga_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
→api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
→http=httplib2.Http())

service = get_unauthorized_service()
data = service.cases().get(body={"case_barcodes": ["TCGA-W5-AA2R","TCGA-ZH-
→A8Y6"]}).execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
→cases
```

{

  "sample_barcodes": ["TCGA-DX-A23Z", "TCGA-B9-7268"

  ]

}

Parameters

Parameter name	Value	Description
case_barcodes	[string]	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "aliquots": [string],
  "clinical_data": {
    "age_at_diagnosis": integer,
    "age_began_smoking_in_years": integer,
    "anatomic_neoplasm_subdivision": string,
    "avg_percent_lymphocyte_infiltration": number,
    "avg_percent_monocyte_infiltration": number,
    "avg_percent_necrosis": number,
```

(continues on next page)

```
        "avg_percent_neutrophil_infiltration": number,
        "avg_percent_normal_cells": number,
        "avg_percent_stromal_cells": number,
        "avg_percent_tumor_cells": number,
        "avg_percent_tumor_nuclei": number,
        "batch_number": integer,
        "bcr": string,
        "bmi": number,
        "case_barcode": string,
        "case_gdc_id": string,
        "clinical_M": string,
        "clinical_N": string,
        "clinical_stage": string,
        "clinical_T": string,
        "colorectal_cancer": string,
        "country": string,
        "days_to_birth": integer,
        "days_to_collection": integer,
        "days_to_death": integer,
        "days_to_initial_pathologic_diagnosis": integer,
        "days_to_last_followup": integer,
        "days_to_last_known_alive": integer,
        "days_to_sample_procurement": integer,
        "days_to_submitted_specimen_dx": integer,
        "disease_code": string,
        "endpoint_type": string,
        "ethnicity": string,
        "gender": string,
        "gleason_score_combined": integer,
        "h_pylori_infection": string,
        "height": integer,
        "histological_type": string,
        "history_of_colon_polyps": string,
        "history_of_neoadjuvant_treatment": string,
        "hpv_calls": string,
        "hpv_status": string,
        "icd_10": string,
        "icd_o_3_histology": string,
        "icd_o_3_site": string,
        "lymphatic_invasion": string,
        "lymphnodes_examined": string,
        "lymphovascular_invasion_present": string,
        "max_percent_lymphocyte_infiltration": number,
        "max_percent_monocyte_infiltration": number,
        "max_percent_necrosis": number,
        "max_percent_neutrophil_infiltration": number,
        "max_percent_normal_cells": number,
        "max_percent_stromal_cells": number,
        "max_percent_tumor_cells": number,
        "max_percent_tumor_nuclei": number,
        "menopause_status": string,
        "min_percent_lymphocyte_infiltration": number,
        "min_percent_monocyte_infiltration": number,
        "min_percent_necrosis": number,
        "min_percent_neutrophil_infiltration": number,
        "min_percent_normal_cells": number,
        "min_percent_stromal_cells": number,
```

```
      "min_percent_tumor_cells": number,
      "min_percent_tumor_nuclei": number,
      "mononucleotide_and_dinucleotide_marker_panel_analysis_status": string,
      "neoplasm_histologic_grade": string,
      "new_tumor_event_after_initial_treatment": string,
      "num_portions": integer,
      "num_slides": integer,
      "number_of_lymphnodes_examined": integer,
      "number_of_lymphnodes_positive_by_he": integer,
      "number_pack_years_smoked": integer,
      "other_dx": string,
      "other_malignancy_anatomic_site": string,
      "other_malignancy_histological_type": string,
      "other_malignancy_type": string,
      "pathologic_M": string,
      "pathologic_N": string,
      "pathologic_stage": string,
      "pathologic_T": string,
      "pathology_report_uuid": string,
      "person_neoplasm_cancer_status": string,
      "pregnancies": string,
      "preservation_method": string,
      "primary_neoplasm_melanoma_dx": string,
      "primary_therapy_outcome_success": string,
      "program_name": string,
      "project_short_name": string,
      "psa_value": number,
      "race": string,
      "residual_tumor": string,
      "sample_barcode": string,
      "sample_gdc_id": string,
      "sample_type": string,
      "stopped_smoking_year": integer,
      "summary_file_count": integer,
      "tobacco_smoking_history": string,
      "tss_code": string,
      "tumor_tissue_site": string,
      "tumor_type": string,
      "venous_invasion": string,
      "vital_status": string,
      "weight": integer,
      "year_of_diagnosis": integer,
      "year_of_tobacco_smoking_onset": integer
   },
   "samples": [string],
   "case_barcode": string
}
```

Parameter name	Value	Description
aliquots[]	list	List of barcodes of aliquots taken from this patient.
clinical_data	nested object	The clinical data about the patient.
clinical_data.age_at_diagnosis	integer	Age at which a condition or disease was first diagnosed in years.

Continued on next page

Table 4 – continued from previous page

Parameter name	Value	Description
clinical_data.age_began_smoking_in_years	integer	Age began smoking cigarettes expressed in number of years since birth.
clinical_data.anatomic_neoplasm_subdivision	string	Text term to describe the spatial location, subdivisions and/or anatomic site name of a tumor.
clinical_data.avg_percent_lymphocyte_infiltration	number	Average in the series of numeric values to represent the percentage of lymphocyte infiltration in a malignant tumor sample or specimen.
clinical_data.avg_percent_monocyte_infiltration	number	Average in the series of numeric values to represent the percentage of monocyte infiltration in a malignant tumor sample or specimen.
clinical_data.avg_percent_necrosis	number	Average in the series of numeric values to represent the percentage of cell death in a malignant tumor sample or specimen.
clinical_data.avg_percent_neutrophil_infiltration	number	Average in the series of numeric values to represent the percentage of neutrophil infiltration in a malignant tumor sample or specimen.
clinical_data.avg_percent_normal_cells	number	Average in the series of numeric values to represent the percentage of normal cells in a malignant tumor sample or specimen.
clinical_data.avg_percent_stromal_cells	number	Average in the series of numeric values to represent the percentage of stromal cells in a malignant tumor sample or specimen.
clinical_data.avg_percent_tumor_cells	number	Average in the series of numeric values to represent the percentage of tumor cells in a malignant tumor sample or specimen.
clinical_data.avg_percent_tumor_nuclei	number	Average in the series of numeric values to represent the percentage of tumor nuclei in a malignant tumor sample or specimen.
clinical_data.batch_number	integer	Groups samples by the batch they were processed in.
clinical_data.bcr	string	A TCGA center where samples are carefully catalogued, processed, quality-checked and stored along with participant clinical information.
clinical_data.bmi	number	Body Mass Index
clinical_data.case_barcode	string	Case barcode.
clinical_data.case_gdc_id	string	The GDC assigned id for the case
clinical_data.clinical_M	string	Extent of the distant metastasis for the cancer based on evidence obtained from clinical assessment parameters determined prior to treatment.
clinical_data.clinical_N	string	Extent of the regional lymph node involvement for the cancer based on evidence obtained from clinical assessment parameters determined prior to treatment.

Continued on next page

Table 4 – continued from previous page

Parameter name	Value	Description
clinical_data.clinical_stage	string	Stage group determined from clinical information on the tumor (T), regional node (N) and metastases (M) and by grouping cases with similar prognosis.
clinical_data.clinical_T	string	Extent of the primary cancer based on evidence obtained from clinical assessment parameters determined prior to treatment.
clinical_data.colorectal_cancer	string	Text term to signify whether a patient has been diagnosed with colorectal cancer.
clinical_data.country	string	Text to identify the name of the state, province, or country in which the sample was procured.
clinical_data.days_to_birth	integer	Time interval from a person's date of birth to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_collection	integer	The number of days between diagnosis and tissue collection.
clinical_data.days_to_death	integer	Time interval from a person's date of death to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_initial_pathologic_diagnosis	integer	Numeric value to represent the day of an individual's initial pathologic diagnosis of cancer.
clinical_data.days_to_last_followup	integer	Time interval from the date of last followup to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_last_known_alive	integer	The number of days between diagnosis and when the individual was last known to be alive.
clinical_data.days_to_sample_procurement	integer	Indicates the days to sample procurement for the submitted sample in relation to the date of initial diagnosis
clinical_data.days_to_submitted_specimen_dx	integer	Time interval from the date of diagnosis of the submitted sample to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.disease_code	string	Text term referring to the cancer type
clinical_data.endpoint_type	string	Which type of GDC Case API was used, either legacy or current
clinical_data.ethnicity	string	The text for reporting information about ethnicity based on the Office of Management and Budget (OMB) categories.
clinical_data.gender	string	Text designations that identify gender.
clinical_data.gleason_score_combined	integer	A numeric value obtained by adding the primary and secondary patterns (grades).
clinical_data.h_pylori_infection	string	Text term to indicate the state of the diagnosis of an individual with Helicobacter pylori infection.
clinical_data.height	integer	The height of the patient in centimeters.

Continued on next page

Table 4 – continued from previous page

Parameter name	Value	Description
clinical_data.histological_type	string	Text term for the structural pattern of cancer cells used to define a microscopic diagnosis.
clinical_data.history_of_colon_polyps	string	Yes/No indicator to describe if the subject had a previous history of colon polyps as noted in the history/physical or previous endoscopic report(s).
clinical_data.history_of_neoadjuvant_treatment	string	Text term to describe the patient's history of neoadjuvant treatment and the kind of treatment given prior to resection of the tumor.
clinical_data.hpv_calls	string	Results of HPV tests.
clinical_data.hpv_status	string	Current HPV status.
clinical_data.icd_10	string	The tenth version of the International Classification of Disease (ICD).
clinical_data.icd_o_3_histology	string	The third edition of the International Classification of Diseases for Oncology.
clinical_data.icd_o_3_site	string	The third edition of the International Classification of Diseases for Oncology.
clinical_data.lymphatic_invasion	string	A yes/no indicator to ask if malignant cells are present in small or thin-walled vessels suggesting lymphatic involvement.
clinical_data.lymphnodes_examined	string	A yes/no/unknown indicator whether a lymph node assessment was performed at the primary presentation of disease.
clinical_data.lymphovascular_invasion_present	string	A yes/no indicator to ask if large vessel (vascular) invasion or small, thin-walled (lymphatic) invasion was detected in a tumor specimen.
clinical_data.max_percent_lymphocyte_infiltration	number	Maximum in the series of numeric values to represent the percentage of lymphocyte infiltration in a malignant tumor sample or specimen.
clinical_data.max_percent_monocyte_infiltration	number	Maximum in the series of numeric values to represent the percentage of monocyte infiltration in a malignant tumor sample or specimen.
clinical_data.max_percent_necrosis	number	Maximum in the series of numeric values to represent the percentage of cell death in a malignant tumor sample or specimen.
clinical_data.max_percent_neutrophil_infiltration	number	Maximum in the series of numeric values to represent the percentage of neutrophil infiltration in a malignant tumor sample or specimen.
clinical_data.max_percent_normal_cells	number	Maximum in the series of numeric values to represent the percentage of normal cells in a malignant tumor sample or specimen.
clinical_data.max_percent_stromal_cells	number	Maximum in the series of numeric values to represent the percentage of stromal cells in a malignant tumor sample or specimen.

Continued on next page

Table 4 – continued from previous page

Parameter name	Value	Description
clinical_data.max_percent_tumor_cells	number	Maximum in the series of numeric values to represent the percentage of tumor cells in a malignant tumor sample or specimen.
clinical_data.max_percent_tumor_nuclei	number	Maximum in the series of numeric values to represent the percentage of tumor nuclei in a malignant tumor sample or specimen.
clinical_data.menopause_status	string	Text term to signify the status of a woman's menopause, the permanent cessation of menses, usually defined by 6 to 12 months of amenorrhea.
clinical_data.min_percent_lymphocyte_infiltration	number	Minimum in the series of numeric values to represent the percentage of lymphcyte infiltration in a malignant tumor sample or specimen.
clinical_data.min_percent_monocyte_infiltration	number	Minimum in the series of numeric values to represent the percentage of monocyte infiltration in a malignant tumor sample or specimen.
clinical_data.min_percent_necrosis	number	Minimum in the series of numeric values to represent the percentage of cell death in a malignant tumor sample or specimen.
clinical_data.min_percent_neutrophil_infiltration	number	Minimum in the series of numeric values to represent the percentage of neutrophil infiltration in a malignant tumor sample or specimen.
clinical_data.min_percent_normal_cells	number	Minimum in the series of numeric values to represent the percentage of normal cells in a malignant tumor sample or specimen.
clinical_data.min_percent_stromal_cells	number	Minimum in the series of numeric values to represent the percentage of stromal cells in a malignant tumor sample or specimen.
clinical_data.min_percent_tumor_cells	number	Minimum in the series of numeric values to represent the percentage of tumor cells in a malignant tumor sample or specimen.
clinical_data.min_percent_tumor_nuclei	number	Minimum in the series of numeric values to represent the percentage of tumor nuclei in a malignant tumor sample or specimen.
clinical_data.mononucleotide_and_dinucleotide_marker_panel_analysis_status	string	The results of microsatellite instability (MSI) testing at using a mononucleotide and dinucleotide microsatellite panel.
clinical_data.neoplasm_histologic_grade	string	Numeric value to express the degree of abnormality of cancer cells, a measure of differentiation and aggressiveness.
clinical_data.new_tumor_event_after_initial_treatment	string	Yes/No/Unknown indicator to identify whether a patient has had a new tumor event after initial treatment.
clinical_data.num_portions	integer	The number of portions obtained from the sample
clinical_data.num_slides	integer	The number of slides derived from the sample

Continued on next page

Table 4 – continued from previous page

Parameter name	Value	Description
clinical_data.number_of_lymphnodes_examined	integer	The total number of lymph nodes removed and pathologically assessed for disease.
clinical_data.number_of_lymphnodes_positive_by_he	integer	Numeric value to signify the count of positive lymph nodes identified through hematoxylin and eosin (H&E) staining light microscopy.
clinical_data.number_pack_years_smoked	integer	Numeric computed value to represent lifetime tobacco exposure defined as number of cigarettes smoked per day x number of years smoked divided by 20.
clinical_data.other_dx	string	Text term to describe the patient's history of cancer diagnosis and the spatial location of any previous cancer occurrence.
clinical_data.other_malignancy_anatomic_site	string	Text term describe the anatomic site of the prior or synchronous malignancy.
clinical_data.other_malignancy_histological_type	string	Text term describe the histology and/or subtype of the prior or synchronous malignancy.
clinical_data.other_malignancy_type	string	The type, relative occurance to the current malignancy
clinical_data.pathologic_M	string	Code to represent the defined absence or presence of distant spread or metastases (M) to locations via vascular channels or lymphatics beyond the regional lymph nodes, using criteria established by the American Joint Committee on Cancer (AJCC).
clinical_data.pathologic_N	string	The codes that represent the stage of cancer based on the nodes present (N stage) according to criteria based on multiple editions of the AJCC's Cancer Staging Manual.
clinical_data.pathologic_stage	string	The extent of a cancer, especially whether the disease has spread from the original site to other parts of the body based on AJCC staging criteria.
clinical_data.pathologic_T	string	Code of pathological T (primary tumor) to define the size or contiguous extension of the primary tumor (T), using staging criteria from the American Joint Committee on Cancer (AJCC).
clinical_data.pathology_report_uuid	string	The UUID of th epathology report
clinical_data.person_neoplasm_cancer_status	string	The state or condition of an individual's neoplasm at a particular point in time.
clinical_data.pregnancies	string	Value to describe the number of full-term pregnancies that a woman has experienced.
clinical_data.preservation_method	string	The method used to preserve the sample after it has been removed from a participant.
clinical_data.primary_neoplasm_melanoma_dx	string	Text indicator to signify whether a person had a primary diagnosis of melanoma.
clinical_data.primary_therapy_outcome_success	string	Measure of success.
clinical_data.program_name	string	Project name, e.g. 'TCGA'.
clinical_data.project_short_name	string	Tumor type abbreviation, e.g. 'BRCA'.

Continued on next page

Table 4 – continued from previous page

Parameter name	Value	Description
clinical_data.psa_value	number	The lab value that represents the results of the most recent (post-operative) prostatic-specific antigen (PSA) in the blood.
clinical_data.race	string	The text for reporting information about race based on the Office of Management and Budget (OMB) categories.
clinical_data.residual_tumor	string	Text terms to describe the status of a tissue margin following surgical resection.
clinical_data.sample_barcode	string	The barcode assigned by TCGA to a sample from a Participant.
clinical_data.sample_gdc_id	string	The GDC assigned id for the sample
clinical_data.sample_type	string	The type of the sample tumor or normal tissue cell or blood sample provided by a participant.
clinical_data.stopped_smoking_year	integer	The year in which the participant quit smoking.
clinical_data.summary_file_count	integer	The count of files associated with the sample
clinical_data.tobacco_smoking_history	string	Category describing current smoking status and smoking history as self-reported by a patient.
clinical_data.tss_code	string	A TSS ID is an alphanumeric code that uniquely identifies a TSS and its associated study
clinical_data.tumor_tissue_site	string	Text term that describes the anatomic site of the tumor or disease.
clinical_data.tumor_type	string	Text term to identify the morphologic subtype of papillary renal cell carcinoma.
clinical_data.venous_invasion	string	The result of an assessment using the Weiss histopathologic criteria.
clinical_data.vital_status	string	The survival state of the person registered on the protocol.
clinical_data.weight	integer	The weight of the patient measured in kilograms.
clinical_data.year_of_diagnosis	integer	Numeric value to represent the year of an individual's initial pathologic diagnosis of cancer.
clinical_data.year_of_tobacco_smoking_onset	integer	The year in which the participant began smoking.
samples[]	list	List of barcodes of samples taken from this patient.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

samples().cloud_storage_file_paths()

Takes a sample barcode as a required parameter and returns cloud storage paths to files associated with that sample.

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
↪samples/TCGA-W5-AA2O-10A/cloud_storage_file_paths
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_tcga_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
↪http=httplib2.Http())

service = get_unauthorized_service()
data = service.samples().cloud_storage_file_paths(sample_barcode='TCGA-W5-
↪AA2R-01A').execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
↪samples/{sample_barcode}/cloud_storage_file_paths
```

Parameters

Parameter name	Value	Description
analysis_workflow_type	string	Optional.
data_category	string	Optional.
data_format	string	Optional.
data_type	string	Optional.
experimental_strategy	string	Optional.
genomic_build	string	Optional.
platform	string	Optional.
sample_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "cloud_storage_file_paths": [string],
  "count": integer
}
```

Parameter name	Value	Description
cloud_storage_file_paths[]	list	List of Google Cloud Storage paths of files associated with the cohort.
count	integer	Number of Google Cloud Storage paths returned for the cohort.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

samples().get()

Given a sample barcode (of length 16, *eg* TCGA-B9-7268-01A), this endpoint returns all available "biospecimen" information about this sample, the associated case barcode, a list of associated aliquots, and a list of "data_details" blocks describing each of the data files associated with this sample

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
↪samples/TCGA-ZH-A8Y6-01A
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_tcga_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
↪http=httplib2.Http())

service = get_unauthorized_service()
data = service.samples().get(sample_barcode='TCGA-W5-AA2R-01A').execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
↪samples/{sample_barcode}
```

Parameters

Parameter name	Value	Description
analysis_workflow_type	string	Optional.
data_category	string	Optional.
data_format	string	Optional.
data_type	string	Optional.
endpoint_type	string	Optional.
experimental_strategy	string	Optional.
platform	string	Optional.
sample_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "aliquots": [string],
  "biospecimen_data": {
    "age_at_diagnosis": integer,
    "age_began_smoking_in_years": integer,
    "anatomic_neoplasm_subdivision": string,
    "avg_percent_lymphocyte_infiltration": number,
    "avg_percent_monocyte_infiltration": number,
    "avg_percent_necrosis": number,
    "avg_percent_neutrophil_infiltration": number,
    "avg_percent_normal_cells": number,
    "avg_percent_stromal_cells": number,
    "avg_percent_tumor_cells": number,
    "avg_percent_tumor_nuclei": number,
    "batch_number": integer,
    "bcr": string,
    "bmi": number,
    "case_barcode": string,
    "case_gdc_id": string,
    "clinical_M": string,
    "clinical_N": string,
    "clinical_stage": string,
    "clinical_T": string,
    "colorectal_cancer": string,
    "country": string,
    "days_to_birth": integer,
    "days_to_collection": integer,
    "days_to_death": integer,
    "days_to_initial_pathologic_diagnosis": integer,
    "days_to_last_followup": integer,
    "days_to_last_known_alive": integer,
    "days_to_sample_procurement": integer,
    "days_to_submitted_specimen_dx": integer,
    "disease_code": string,
    "endpoint_type": string,
    "ethnicity": string,
    "gender": string,
    "gleason_score_combined": integer,
    "h_pylori_infection": string,
    "height": integer,
    "histological_type": string,
    "history_of_colon_polyps": string,
```

(continues on next page)

```
    "history_of_neoadjuvant_treatment": string,
    "hpv_calls": string,
    "hpv_status": string,
    "icd_10": string,
    "icd_o_3_histology": string,
    "icd_o_3_site": string,
    "lymphatic_invasion": string,
    "lymphnodes_examined": string,
    "lymphovascular_invasion_present": string,
    "max_percent_lymphocyte_infiltration": number,
    "max_percent_monocyte_infiltration": number,
    "max_percent_necrosis": number,
    "max_percent_neutrophil_infiltration": number,
    "max_percent_normal_cells": number,
    "max_percent_stromal_cells": number,
    "max_percent_tumor_cells": number,
    "max_percent_tumor_nuclei": number,
    "menopause_status": string,
    "min_percent_lymphocyte_infiltration": number,
    "min_percent_monocyte_infiltration": number,
    "min_percent_necrosis": number,
    "min_percent_neutrophil_infiltration": number,
    "min_percent_normal_cells": number,
    "min_percent_stromal_cells": number,
    "min_percent_tumor_cells": number,
    "min_percent_tumor_nuclei": number,
    "mononucleotide_and_dinucleotide_marker_panel_analysis_status": string,
    "neoplasm_histologic_grade": string,
    "new_tumor_event_after_initial_treatment": string,
    "num_portions": integer,
    "num_slides": integer,
    "number_of_lymphnodes_examined": integer,
    "number_of_lymphnodes_positive_by_he": integer,
    "number_pack_years_smoked": integer,
    "other_dx": string,
    "other_malignancy_anatomic_site": string,
    "other_malignancy_histological_type": string,
    "other_malignancy_type": string,
    "pathologic_M": string,
    "pathologic_N": string,
    "pathologic_stage": string,
    "pathologic_T": string,
    "pathology_report_uuid": string,
    "person_neoplasm_cancer_status": string,
    "pregnancies": string,
    "preservation_method": string,
    "primary_neoplasm_melanoma_dx": string,
    "primary_therapy_outcome_success": string,
    "program_name": string,
    "project_short_name": string,
    "psa_value": number,
    "race": string,
    "residual_tumor": string,
    "sample_barcode": string,
    "sample_gdc_id": string,
    "sample_type": string,
    "stopped_smoking_year": integer,
```

```
      "summary_file_count": integer,
      "tobacco_smoking_history": string,
      "tss_code": string,
      "tumor_tissue_site": string,
      "tumor_type": string,
      "venous_invasion": string,
      "vital_status": string,
      "weight": integer,
      "year_of_diagnosis": integer,
      "year_of_tobacco_smoking_onset": integer
   },
   "case_barcode": string,
   "case_gdc_id": string,
   "data_details": [
      {
         "access": string,
         "analysis_workflow_type": string,
         "data_category": string,
         "data_format": string,
         "data_type": string,
         "disease_code": string,
         "endpoint_type": string,
         "experimental_strategy": string,
         "file_gdc_id": string,
         "file_name": string,
         "file_name_key": string,
         "file_size": string,
         "index_file_name": string,
         "platform": string,
         "program_name": string,
         "project_short_name": string,
         "sample_barcode": string,
         "sample_gdc_id": string,
         "sample_type": string
      }
   ],
   "data_details_count": integer
}
```

aliquots[]	list	List of barcodes of aliquots tak
biospecimen_data	nested object	Biospecimen data about the sa
biospecimen_data.age_at_diagnosis	integer	Age at which a condition or di
biospecimen_data.age_began_smoking_in_years	integer	Age began smoking cigarettes
biospecimen_data.anatomic_neoplasm_subdivision	string	Text term to describe the spati
biospecimen_data.avg_percent_lymphocyte_infiltration	number	Average in the series of numer
biospecimen_data.avg_percent_monocyte_infiltration	number	Average in the series of numer
biospecimen_data.avg_percent_necrosis	number	Average in the series of numer
biospecimen_data.avg_percent_neutrophil_infiltration	number	Average in the series of numer
biospecimen_data.avg_percent_normal_cells	number	Average in the series of numer
biospecimen_data.avg_percent_stromal_cells	number	Average in the series of numer
biospecimen_data.avg_percent_tumor_cells	number	Average in the series of numer

biospecimen_data.avg_percent_tumor_nuclei	number	Average in the series of numer
biospecimen_data.batch_number	integer	Groups samples by the batch t
biospecimen_data.bcr	string	A TCGA center where sample
biospecimen_data.bmi	number	Body Mass Index
biospecimen_data.case_barcode	string	Case barcode.
biospecimen_data.case_gdc_id	string	The GDC assigned id for the c
biospecimen_data.clinical_M	string	Extent of the distant metastasi
biospecimen_data.clinical_N	string	Extent of the regional lymph r
biospecimen_data.clinical_stage	string	Stage group determined from c
biospecimen_data.clinical_T	string	Extent of the primary cancer b
biospecimen_data.colorectal_cancer	string	Text term to signify whether a
biospecimen_data.country	string	Text to identify the name of th
biospecimen_data.days_to_birth	integer	Time interval from a person's
biospecimen_data.days_to_collection	integer	The number of days between c
biospecimen_data.days_to_death	integer	Time interval from a person's
biospecimen_data.days_to_initial_pathologic_diagnosis	integer	Numeric value to represent the
biospecimen_data.days_to_last_followup	integer	Time interval from the date of
biospecimen_data.days_to_last_known_alive	integer	The number of days between c
biospecimen_data.days_to_sample_procurement	integer	Indicates the days to sample p
biospecimen_data.days_to_submitted_specimen_dx	integer	Time interval from the date of
biospecimen_data.disease_code	string	Text term referring to the canc
biospecimen_data.endpoint_type	string	Which type of GDC Case API
biospecimen_data.ethnicity	string	The text for reporting informa
biospecimen_data.gender	string	Text designations that identify
biospecimen_data.gleason_score_combined	integer	A numeric value obtained by a
biospecimen_data.h_pylori_infection	string	Text term to indicate the state
biospecimen_data.height	integer	The height of the patient in ce
biospecimen_data.histological_type	string	Text term for the structural pa
biospecimen_data.history_of_colon_polyps	string	Yes/No indicator to describe if
biospecimen_data.history_of_neoadjuvant_treatment	string	Text term to describe the patie
biospecimen_data.hpv_calls	string	Results of HPV tests.
biospecimen_data.hpv_status	string	Current HPV status.
biospecimen_data.icd_10	string	The tenth version of the Intern
biospecimen_data.icd_o_3_histology	string	The third edition of the Interna
biospecimen_data.icd_o_3_site	string	The third edition of the Interna
biospecimen_data.lymphatic_invasion	string	A yes/no indicator to ask if ma
biospecimen_data.lymphnodes_examined	string	A yes/no/unknown indicator w
biospecimen_data.lymphovascular_invasion_present	string	A yes/no indicator to ask if lan
biospecimen_data.max_percent_lymphocyte_infiltration	number	Maximum in the series of num
biospecimen_data.max_percent_monocyte_infiltration	number	Maximum in the series of num
biospecimen_data.max_percent_necrosis	number	Maximum in the series of num
biospecimen_data.max_percent_neutrophil_infiltration	number	Maximum in the series of num
biospecimen_data.max_percent_normal_cells	number	Maximum in the series of num
biospecimen_data.max_percent_stromal_cells	number	Maximum in the series of num
biospecimen_data.max_percent_tumor_cells	number	Maximum in the series of num
biospecimen_data.max_percent_tumor_nuclei	number	Maximum in the series of num
biospecimen_data.menopause_status	string	Text term to signify the status
biospecimen_data.min_percent_lymphocyte_infiltration	number	Minimum in the series of num
biospecimen_data.min_percent_monocyte_infiltration	number	Minimum in the series of num
biospecimen_data.min_percent_necrosis	number	Minimum in the series of num

biospecimen_data.min_percent_neutrophil_infiltration	number	Minimum in the series of num
biospecimen_data.min_percent_normal_cells	number	Minimum in the series of num
biospecimen_data.min_percent_stromal_cells	number	Minimum in the series of num
biospecimen_data.min_percent_tumor_cells	number	Minimum in the series of num
biospecimen_data.min_percent_tumor_nuclei	number	Minimum in the series of num
biospecimen_data.mononucleotide_and_dinucleotide_marker_panel_analysis_status	string	Text result of microsatellite in
biospecimen_data.neoplasm_histologic_grade	string	Numeric value to express the c
biospecimen_data.new_tumor_event_after_initial_treatment	string	Yes/No/Unknown indicator to
biospecimen_data.num_portions	integer	The number of portions obtain
biospecimen_data.num_slides	integer	The number of slides derived f
biospecimen_data.number_of_lymphnodes_examined	integer	The total number of lymph no
biospecimen_data.number_of_lymphnodes_positive_by_he	integer	Numeric value to signify the c
biospecimen_data.number_pack_years_smoked	integer	Numeric computed value to re
biospecimen_data.other_dx	string	Text term to describe the patie
biospecimen_data.other_malignancy_anatomic_site	string	Text term describe the anatom
biospecimen_data.other_malignancy_histological_type	string	Text term describe the histolog
biospecimen_data.other_malignancy_type	string	The type, relative occurance to
biospecimen_data.pathologic_M	string	Code to represent the defined a
biospecimen_data.pathologic_N	string	The codes that represent the st
biospecimen_data.pathologic_stage	string	The extent of a cancer, especia
biospecimen_data.pathologic_T	string	Code of pathological T (prima
biospecimen_data.pathology_report_uuid	string	The UUID of th epathology re
biospecimen_data.person_neoplasm_cancer_status	string	The state or condition of an in
biospecimen_data.pregnancies	string	Value to describe the number c
biospecimen_data.preservation_method	string	The method used to preserve t
biospecimen_data.primary_neoplasm_melanoma_dx	string	Text indicator to signify wheth
biospecimen_data.primary_therapy_outcome_success	string	Measure of success.
biospecimen_data.program_name	string	Project name, e.g. 'TCGA'.
biospecimen_data.project_short_name	string	Tumor type abbreviation, e.g.
biospecimen_data.psa_value	number	The lab value that represents t
biospecimen_data.race	string	The text for reporting informa
biospecimen_data.residual_tumor	string	Text terms to describe the statu
biospecimen_data.sample_barcode	string	The barcode assigned by TCG
biospecimen_data.sample_gdc_id	string	The GDC assigned id for the s
biospecimen_data.sample_type	string	The type of the sample tumor
biospecimen_data.stopped_smoking_year	integer	The year in which the particip
biospecimen_data.summary_file_count	integer	The count of files associated w
biospecimen_data.tobacco_smoking_history	string	Category describing current sm
biospecimen_data.tss_code	string	A TSS ID is an alphanumeric
biospecimen_data.tumor_tissue_site	string	Text term that describes the an
biospecimen_data.tumor_type	string	Text term to identify the morp
biospecimen_data.venous_invasion	string	The result of an assessment us
biospecimen_data.vital_status	string	The survival state of the perso
biospecimen_data.weight	integer	The weight of the patient mea
biospecimen_data.year_of_diagnosis	integer	Numeric value to represent the
biospecimen_data.year_of_tobacco_smoking_onset	integer	The year in which the particip
case_barcode	string	Case barcode.
case_gdc_id	string	Case gdc id.
data_details[]	list	List of information about each
data_details[].access	string	An indication of the security p

data_details[].analysis_workflow_type	string	The type of workflow used to
data_details[].data_category	string	The higher level categorization
data_details[].data_format	string	The format of the data file, e.g
data_details[].data_type	string	Data type stored in Google Cl
data_details[].disease_code	string	The disease abbeviation, e.g. '
data_details[].endpoint_type	string	The GDC files API the data fil
data_details[].experimental_strategy	string	The sequencing, array or other
data_details[].file_gdc_id	string	The GDC assigned id for the f
data_details[].file_name	string	Name of the datafile stored on
data_details[].file_name_key	string	Google Cloud Storage path to
data_details[].file_size	string	The size of the file
data_details[].index_file_name	string	For BAM files, the name of its
data_details[].platform	string	The sequencing or array platfo
data_details[].program_name	string	The program for which the da
data_details[].project_short_name	string	The id of the project, e.g. 'CC
data_details[].sample_barcode	string	Sample barcode.
data_details[].sample_gdc_id	string	The GDC assigned id for the s
data_details[].sample_type	string	The sample type, e.g. '01', '10
data_details_count	integer	Number of files associated wit

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

samples().get_list()

Given a list of sample barcodes (of length 16, *eg* TCGA-B9-7268-01A), this endpoint returns all available "biospecimen" information about these samples, the associated case barcode, a list of associated aliquots, and a list of "data_details" blocks describing each of the data files associated with these samples

Example:

```
curl --data '{"sample_barcodes": ["TCGA-DX-A23Z-10A","TCGA-B9-7268-01A"]}'
→https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
→samples
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_tcga_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
→api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
→http=httplib2.Http())
```

```
service = get_unauthorized_service()
data = service.samples().get(body={"sample_barcodes": ["TCGA-W5-AA2R-01A",
→"TCGA-DX-A23Z-10A"]}).execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
→samples
```

{

> **"sample_barcodes": [** "TCGA-DX-A23Z-10A", "TCGA-B9-7268-01A"
>
> **]**

}

Parameters

Parameter name	Value	Description
analysis_workflow_type	string	Optional.
data_category	string	Optional.
data_format	string	Optional.
data_type	string	Optional.
endpoint_type	string	Optional.
experimental_strategy	string	Optional.
platform	string	Optional.
sample_barcodes	[string]	Required.

Response

If successful, this method returns a response body with the following structure:

```
[
  {
    "aliquots": [string],
    "biospecimen_data": {
      "age_at_diagnosis": integer,
      "age_began_smoking_in_years": integer,
      "anatomic_neoplasm_subdivision": string,
      "avg_percent_lymphocyte_infiltration": number,
      "avg_percent_monocyte_infiltration": number,
      "avg_percent_necrosis": number,
      "avg_percent_neutrophil_infiltration": number,
      "avg_percent_normal_cells": number,
      "avg_percent_stromal_cells": number,
      "avg_percent_tumor_cells": number,
      "avg_percent_tumor_nuclei": number,
      "batch_number": integer,
      "bcr": string,
      "bmi": number,
      "case_barcode": string,
      "case_gdc_id": string,
      "clinical_M": string,
      "clinical_N": string,
```

```
        "clinical_stage": string,
        "clinical_T": string,
        "colorectal_cancer": string,
        "country": string,
        "days_to_birth": integer,
        "days_to_collection": integer,
        "days_to_death": integer,
        "days_to_initial_pathologic_diagnosis": integer,
        "days_to_last_followup": integer,
        "days_to_last_known_alive": integer,
        "days_to_sample_procurement": integer,
        "days_to_submitted_specimen_dx": integer,
        "disease_code": string,
        "endpoint_type": string,
        "ethnicity": string,
        "gender": string,
        "gleason_score_combined": integer,
        "h_pylori_infection": string,
        "height": integer,
        "histological_type": string,
        "history_of_colon_polyps": string,
        "history_of_neoadjuvant_treatment": string,
        "hpv_calls": string,
        "hpv_status": string,
        "icd_10": string,
        "icd_o_3_histology": string,
        "icd_o_3_site": string,
        "lymphatic_invasion": string,
        "lymphnodes_examined": string,
        "lymphovascular_invasion_present": string,
        "max_percent_lymphocyte_infiltration": number,
        "max_percent_monocyte_infiltration": number,
        "max_percent_necrosis": number,
        "max_percent_neutrophil_infiltration": number,
        "max_percent_normal_cells": number,
        "max_percent_stromal_cells": number,
        "max_percent_tumor_cells": number,
        "max_percent_tumor_nuclei": number,
        "menopause_status": string,
        "min_percent_lymphocyte_infiltration": number,
        "min_percent_monocyte_infiltration": number,
        "min_percent_necrosis": number,
        "min_percent_neutrophil_infiltration": number,
        "min_percent_normal_cells": number,
        "min_percent_stromal_cells": number,
        "min_percent_tumor_cells": number,
        "min_percent_tumor_nuclei": number,
        "mononucleotide_and_dinucleotide_marker_panel_analysis_status":␣
↪string,
        "neoplasm_histologic_grade": string,
        "new_tumor_event_after_initial_treatment": string,
        "num_portions": integer,
        "num_slides": integer,
        "number_of_lymphnodes_examined": integer,
        "number_of_lymphnodes_positive_by_he": integer,
        "number_pack_years_smoked": integer,
        "other_dx": string,
```

```
      "other_malignancy_anatomic_site": string,
      "other_malignancy_histological_type": string,
      "other_malignancy_type": string,
      "pathologic_M": string,
      "pathologic_N": string,
      "pathologic_stage": string,
      "pathologic_T": string,
      "pathology_report_uuid": string,
      "person_neoplasm_cancer_status": string,
      "pregnancies": string,
      "preservation_method": string,
      "primary_neoplasm_melanoma_dx": string,
      "primary_therapy_outcome_success": string,
      "program_name": string,
      "project_short_name": string,
      "psa_value": number,
      "race": string,
      "residual_tumor": string,
      "sample_barcode": string,
      "sample_gdc_id": string,
      "sample_type": string,
      "stopped_smoking_year": integer,
      "summary_file_count": integer,
      "tobacco_smoking_history": string,
      "tss_code": string,
      "tumor_tissue_site": string,
      "tumor_type": string,
      "venous_invasion": string,
      "vital_status": string,
      "weight": integer,
      "year_of_diagnosis": integer,
      "year_of_tobacco_smoking_onset": integer
   },
   "case_barcode": string,
   "case_gdc_id": string,
   "data_details": [
      {
         "access": string,
         "analysis_workflow_type": string,
         "data_category": string,
         "data_format": string,
         "data_type": string,
         "disease_code": string,
         "endpoint_type": string,
         "experimental_strategy": string,
         "file_gdc_id": string,
         "file_name": string,
         "file_name_key": string,
         "file_size": string,
         "index_file_name": string,
         "platform": string,
         "program_name": string,
         "project_short_name": string,
         "sample_barcode": string,
         "sample_gdc_id": string,
         "sample_type": string
      }
```

```
    ],
    "data_details_count": integer,
    "sample_barcode": string
  }
]
```

aliquots[]	list	List of barcodes of aliquots ta
biospecimen_data	nested object	Biospecimen data about the sa
biospecimen_data.age_at_diagnosis	integer	Age at which a condition or di
biospecimen_data.age_began_smoking_in_years	integer	Age began smoking cigarettes
biospecimen_data.anatomic_neoplasm_subdivision	string	Text term to describe the spati
biospecimen_data.avg_percent_lymphocyte_infiltration	number	Average in the series of numer
biospecimen_data.avg_percent_monocyte_infiltration	number	Average in the series of numer
biospecimen_data.avg_percent_necrosis	number	Average in the series of numer
biospecimen_data.avg_percent_neutrophil_infiltration	number	Average in the series of numer
biospecimen_data.avg_percent_normal_cells	number	Average in the series of numer
biospecimen_data.avg_percent_stromal_cells	number	Average in the series of numer
biospecimen_data.avg_percent_tumor_cells	number	Average in the series of numer
biospecimen_data.avg_percent_tumor_nuclei	number	Average in the series of numer
biospecimen_data.batch_number	integer	Groups samples by the batch t
biospecimen_data.bcr	string	A TCGA center where sample
biospecimen_data.bmi	number	Body Mass Index
biospecimen_data.case_barcode	string	Case barcode.
biospecimen_data.case_gdc_id	string	The GDC assigned id for the c
biospecimen_data.clinical_M	string	Extent of the distant metastasi
biospecimen_data.clinical_N	string	Extent of the regional lymph r
biospecimen_data.clinical_stage	string	Stage group determined from c
biospecimen_data.clinical_T	string	Extent of the primary cancer b
biospecimen_data.colorectal_cancer	string	Text term to signify whether a
biospecimen_data.country	string	Text to identify the name of th
biospecimen_data.days_to_birth	integer	Time interval from a person's
biospecimen_data.days_to_collection	integer	The number of days between c
biospecimen_data.days_to_death	integer	Time interval from a person's
biospecimen_data.days_to_initial_pathologic_diagnosis	integer	Numeric value to represent the
biospecimen_data.days_to_last_followup	integer	Time interval from the date of
biospecimen_data.days_to_last_known_alive	integer	The number of days between c
biospecimen_data.days_to_sample_procurement	integer	Indicates the days to sample p
biospecimen_data.days_to_submitted_specimen_dx	integer	Time interval from the date of
biospecimen_data.disease_code	string	Text term referring to the canc
biospecimen_data.endpoint_type	string	Which type of GDC Case API
biospecimen_data.ethnicity	string	The text for reporting informa
biospecimen_data.gender	string	Text designations that identify
biospecimen_data.gleason_score_combined	integer	A numeric value obtained by a
biospecimen_data.h_pylori_infection	string	Text term to indicate the state
biospecimen_data.height	integer	The height of the patient in ce
biospecimen_data.histological_type	string	Text term for the structural pat
biospecimen_data.history_of_colon_polyps	string	Yes/No indicator to describe if
biospecimen_data.history_of_neoadjuvant_treatment	string	Text term to describe the patie

biospecimen_data.hpv_calls	string	Results of HPV tests.
biospecimen_data.hpv_status	string	Current HPV status.
biospecimen_data.icd_10	string	The tenth version of the Interna
biospecimen_data.icd_o_3_histology	string	The third edition of the Interna
biospecimen_data.icd_o_3_site	string	The third edition of the Interna
biospecimen_data.lymphatic_invasion	string	A yes/no indicator to ask if ma
biospecimen_data.lymphnodes_examined	string	A yes/no/unknown indicator w
biospecimen_data.lymphovascular_invasion_present	string	A yes/no indicator to ask if lan
biospecimen_data.max_percent_lymphocyte_infiltration	number	Maximum in the series of num
biospecimen_data.max_percent_monocyte_infiltration	number	Maximum in the series of num
biospecimen_data.max_percent_necrosis	number	Maximum in the series of num
biospecimen_data.max_percent_neutrophil_infiltration	number	Maximum in the series of num
biospecimen_data.max_percent_normal_cells	number	Maximum in the series of num
biospecimen_data.max_percent_stromal_cells	number	Maximum in the series of num
biospecimen_data.max_percent_tumor_cells	number	Maximum in the series of num
biospecimen_data.max_percent_tumor_nuclei	number	Maximum in the series of num
biospecimen_data.menopause_status	string	Text term to signify the status
biospecimen_data.min_percent_lymphocyte_infiltration	number	Minimum in the series of num
biospecimen_data.min_percent_monocyte_infiltration	number	Minimum in the series of num
biospecimen_data.min_percent_necrosis	number	Minimum in the series of num
biospecimen_data.min_percent_neutrophil_infiltration	number	Minimum in the series of num
biospecimen_data.min_percent_normal_cells	number	Minimum in the series of num
biospecimen_data.min_percent_stromal_cells	number	Minimum in the series of num
biospecimen_data.min_percent_tumor_cells	number	Minimum in the series of num
biospecimen_data.min_percent_tumor_nuclei	number	Minimum in the series of num
biospecimen_data.mononucleotide_and_dinucleotide_marker_panel_analysis_status	string	Text result of microsatellite in
biospecimen_data.neoplasm_histologic_grade	string	Numeric value to express the c
biospecimen_data.new_tumor_event_after_initial_treatment	string	Yes/No/Unknown indicator to
biospecimen_data.num_portions	integer	The number of portions obtai
biospecimen_data.num_slides	integer	The number of slides derived f
biospecimen_data.number_of_lymphnodes_examined	integer	The total number of lymph no
biospecimen_data.number_of_lymphnodes_positive_by_he	integer	Numeric value to signify the c
biospecimen_data.number_pack_years_smoked	integer	Numeric computed value to re
biospecimen_data.other_dx	string	Text term to describe the patie
biospecimen_data.other_malignancy_anatomic_site	string	Text term describe the anatom
biospecimen_data.other_malignancy_histological_type	string	Text term describe the histolog
biospecimen_data.other_malignancy_type	string	The type, relative occurance to
biospecimen_data.pathologic_M	string	Code to represent the defined a
biospecimen_data.pathologic_N	string	The codes that represent the st
biospecimen_data.pathologic_stage	string	The extent of a cancer, especia
biospecimen_data.pathologic_T	string	Code of pathological T (prima
biospecimen_data.pathology_report_uuid	string	The UUID of th epathology re
biospecimen_data.person_neoplasm_cancer_status	string	The state or condition of an in
biospecimen_data.pregnancies	string	Value to describe the number c
biospecimen_data.preservation_method	string	The method used to preserve t
biospecimen_data.primary_neoplasm_melanoma_dx	string	Text indicator to signify wheth
biospecimen_data.primary_therapy_outcome_success	string	Measure of success.
biospecimen_data.program_name	string	Project name, e.g. 'TCGA'.
biospecimen_data.project_short_name	string	Tumor type abbreviation, e.g.
biospecimen_data.psa_value	number	The lab value that represents t

biospecimen_data.race	string	The text for reporting informa
biospecimen_data.residual_tumor	string	Text terms to describe the statu
biospecimen_data.sample_barcode	string	The barcode assigned by TCG
biospecimen_data.sample_gdc_id	string	The GDC assigned id for the s
biospecimen_data.sample_type	string	The type of the sample tumor
biospecimen_data.stopped_smoking_year	integer	The year in which the particip
biospecimen_data.summary_file_count	integer	The count of files associated w
biospecimen_data.tobacco_smoking_history	string	Category describing current sr
biospecimen_data.tss_code	string	A TSS ID is an alphanumeric
biospecimen_data.tumor_tissue_site	string	Text term that describes the ar
biospecimen_data.tumor_type	string	Text term to identify the morp
biospecimen_data.venous_invasion	string	The result of an assessment us
biospecimen_data.vital_status	string	The survival state of the perso
biospecimen_data.weight	integer	The weight of the patient mea
biospecimen_data.year_of_diagnosis	integer	Numeric value to represent the
biospecimen_data.year_of_tobacco_smoking_onset	integer	The year in which the particip
case_barcode	string	Case barcode.
case_gdc_id	string	Case gdc id.
data_details[]	list	List of information about each
data_details[].access	string	An indication of the security p
data_details[].analysis_workflow_type	string	The type of workflow used to
data_details[].data_category	string	The higher level categorizatio
data_details[].data_format	string	The format of the data file, e.g
data_details[].data_type	string	Data type stored in Google Cl
data_details[].disease_code	string	The disease abbeviation, e.g. '
data_details[].endpoint_type	string	The GDC files API the data fil
data_details[].experimental_strategy	string	The sequencing, array or other
data_details[].file_gdc_id	string	The GDC assigned id for the f
data_details[].file_name	string	Name of the datafile stored on
data_details[].file_name_key	string	Google Cloud Storage path to
data_details[].file_size	string	The size of the file
data_details[].index_file_name	string	For BAM files, the name of its
data_details[].platform	string	The sequencing or array platfo
data_details[].program_name	string	The program for which the da
data_details[].project_short_name	string	The id of the project, e.g. 'CC
data_details[].sample_barcode	string	Sample barcode.
data_details[].sample_gdc_id	string	The GDC assigned id for the s
data_details[].sample_type	string	The sample type, e.g. '01', '10
data_details_count	integer	Number of files associated wit

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

users().get()

Returns the dbGaP authorization status of the user.

Example:

```
python isb_curl.py https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_
↪api/v3/tcga/users
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
from oauth2client.client import OAuth2WebServerFlow
from oauth2client import tools
from oauth2client.file import Storage
import httplib2
import os

CLIENT_ID = '907668440978-0ol0griu70qkeb6k3gnn2vipfa5mgl60.apps.
→googleusercontent.com'
CLIENT_SECRET = 'To_WJH7-1V-TofhNGcEqmEYi'
EMAIL_SCOPE = 'https://www.googleapis.com/auth/userinfo.email'
DEFAULT_STORAGE_FILE = os.path.join(os.path.expanduser('~'), '.isb_
→credentials')


def get_credentials():
        oauth_flow_args = ['--noauth_local_webserver']
        storage = Storage(DEFAULT_STORAGE_FILE)
        credentials = storage.get()
        if not credentials or credentials.invalid:
                flow = OAuth2WebServerFlow(CLIENT_ID, CLIENT_SECRET, EMAIL_
→SCOPE)
                flow.auth_uri = flow.auth_uri.rstrip('/') + '?approval_
→prompt=force'
                credentials = tools.run_flow(flow, storage, tools.argparser.
→parse_args(oauth_flow_args))
        return credentials


def get_authorized_service():
        api = 'isb_cgc_tcga_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
→api, version)
        credentials = get_credentials()
        http = credentials.authorize(httplib2.Http())
        if credentials.access_token_expired or credentials.invalid:
                credentials.refresh(http)
        authorized_service = build(api, version,
→discoveryServiceUrl=discovery_url, http=http)
        return authorized_service


service = get_authorized_service()
data = service.users().get().execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/tcga/
→users
```

Parameters

None

Response

If successful, this method returns a response body with the following structure:

```
{
  "dbGaP_allowed": boolean,
  "dbGaP_authorized": boolean,
  "message": string
}
```

Parameter name	Value	Description
dbGaP_authorized	boolean	True or false.
message	string	Message indicating the authorization status of the user.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

aliquots().annotations()

Returns TCGA annotations about a specific aliquot, Takes an aliquot barcode (of length 28, *eg* TCGA-01-0628-11A-01D-0356-01) as a required parameter. User does not need to be authenticated.

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/
↪aliquots/TCGA-01-0628-11A-01D-0358-06/annotations
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_tcga_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
↪http=httplib2.Http())

service = get_unauthorized_service()
data = service.aliquots().annotations(aliquot_barcode='TCGA-01-0628-11A-01D-
↪0358-06').execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/aliquots/
↪{aliquot_barcode}/annotations
```

Parameters

Parameter name	Value	Description
aliquot_barcode	string	Required.
entity_type	string	Optional.

Response

If successful, this method returns a response body with the following structure:

```
{
  "count": integer,
  "items": [
    {
      "aliquot_barcode": string,
      "annotation_gdc_id": string,
      "annotation_submitter_id": string,
      "case_barcode": string,
      "case_gdc_id": string,
      "category": string,
      "classification": string,
      "endpoint_type": string,
      "entity_barcode": string,
      "entity_gdc_id": string,
      "entity_type": string,
      "notes": string,
      "program_name": string,
      "project_short_name": string,
      "sample_barcode": string,
      "status": string
    }
  ]
}
```

Parameter name	Value	Description
count	integer	Number of annotations returned.
items[]	list	List of annotation items.
items[].aliquot_barcode	string	Aliquot barcode.
items[].annotation_gdc_id	string	Id assigned by the GDC to the annotation
items[].annotation_submitter_id	string	Id assigned to the annotation by the TCGA
items[].case_barcode	string	Case barcode.
items[].case_gdc_id	string	Id assigned by the GDC to the case
items[].category	string	Annotation category name, e.g. 'Acceptable treatment for TCGA tumor'.
items[].classification	string	Annotation classification, .e.g 'CenterNotification', 'Notification', 'Observation', or 'Redaction'.
items[].endpoint_type	string	Which type of GDC Annotation API was used, either legacy or current
items[].entity_barcode	string	The TCGA barcode that the annottion is associated with
items[].entity_gdc_id	string	Id assigned by the GDC to the entity
items[].entity_type	string	Entity type, e.g. 'Case', 'Aliquot', 'Analyte', 'Portion'', 'Slide', or 'Sample'.
items[].notes	string	Notes on the annotation
items[].program_name	string	The program name, e.g. 'TCGA' (the only program with annotations)
items[].project_short_name	string	The project id, e.g. 'TCGA-BRCA', 'TCGA-OV'.
items[].sample_barcode	string	Sample barcode.
items[].status	string	Status of the annotation, e.g. 'Approved', 'Rescinded'

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cases().annotations()

Returns TCGA annotations about a specific sample, Takes a case barcode (of length 12, *eg* TCGA-01-0628) as a required parameter. User does not need to be authenticated.

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/cases/
↪TCGA-01-0628/annotations
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```
from googleapiclient.discovery import build
import httplib2


def get_unauthorized_service():
        api = 'isb_cgc_tcga_api'
```

(continues on next page)

```
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
→api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
→http=httplib2.Http())

service = get_unauthorized_service()
data = service.cases().annotations(sample_barcode='TCGA-01-0628').execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/cases/
→{case_barcode}/annotations
```

Parameters

Parameter name	Value	Description
case_barcode	string	Required.
entity_type	string	Optional.

Response

If successful, this method returns a response body with the following structure:

```
{
  "count": integer,
  "items": [
    {
      "aliquot_barcode": string,
      "annotation_gdc_id": string,
      "annotation_submitter_id": string,
      "case_barcode": string,
      "case_gdc_id": string,
      "category": string,
      "classification": string,
      "endpoint_type": string,
      "entity_barcode": string,
      "entity_gdc_id": string,
      "entity_type": string,
      "notes": string,
      "program_name": string,
      "project_short_name": string,
      "sample_barcode": string,
      "status": string
    }
  ]
}
```

Parameter name	Value	Description
count	integer	Number of annotations returned.
items[]	list	List of annotation items.
items[].aliquot_barcode	string	Aliquot barcode.
items[].annotation_gdc_id	string	Id assigned by the GDC to the annotation
items[].annotation_submitter_id	string	Id assigned to the annotation by the TCGA
items[].case_barcode	string	Case barcode.
items[].case_gdc_id	string	Id assigned by the GDC to the case
items[].category	string	Annotation category name, e.g. 'Acceptable treatment for TCGA tumor'.
items[].classification	string	Annotation classification, .e.g 'CenterNotification', 'Notification', 'Observation', or 'Redaction'.
items[].endpoint_type	string	Which type of GDC Annotation API was used, either legacy or current
items[].entity_barcode	string	The TCGA barcode that the annottion is associated with
items[].entity_gdc_id	string	Id assigned by the GDC to the entity
items[].entity_type	string	Entity type, e.g. 'Case', 'Aliquot', 'Analyte', 'Portion'', 'Slide', or 'Sample'.
items[].notes	string	Notes on the annotation
items[].program_name	string	The program name, e.g. 'TCGA' (the only program with annotations)
items[].project_short_name	string	The project id, e.g. 'TCGA-BRCA', 'TCGA-OV'.
items[].sample_barcode	string	Sample barcode.
items[].status	string	Status of the annotation, e.g. 'Approved', 'Rescinded'

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

samples().annotations()

Returns TCGA annotations about a specific sample, Takes a sample barcode (of length 16, *eg* TCGA-01-0628-11A) as a required parameter. User does not need to be authenticated.

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/samples/
↪TCGA-01-0628-11A/annotations
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2


def get_unauthorized_service():
        api = 'isb_cgc_tcga_api'
```

(continues on next page)

```
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
→api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
→http=httplib2.Http())

service = get_unauthorized_service()
data = service.samples().annotations(sample_barcode='TCGA-01-0628-11A').
→execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_tcga_api/v3/samples/
→{sample_barcode}/annotations
```

Parameters

Parameter name	Value	Description
entity_type	string	Optional.
sample_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "count": integer,
  "items": [
    {
      "aliquot_barcode": string,
      "annotation_gdc_id": string,
      "annotation_submitter_id": string,
      "case_barcode": string,
      "case_gdc_id": string,
      "category": string,
      "classification": string,
      "endpoint_type": string,
      "entity_barcode": string,
      "entity_gdc_id": string,
      "entity_type": string,
      "notes": string,
      "program_name": string,
      "project_short_name": string,
      "sample_barcode": string,
      "status": string
    }
  ]
}
```

Parameter name	Value	Description
count	integer	Number of annotations returned.
items[]	list	List of annotation items.
items[].aliquot_barcode	string	Aliquot barcode.
items[].annotation_gdc_id	string	Id assigned by the GDC to the annotation
items[].annotation_submitter_id	string	Id assigned to the annotation by the TCGA
items[].case_barcode	string	Case barcode.
items[].case_gdc_id	string	Id assigned by the GDC to the case
items[].category	string	Annotation category name, e.g. 'Acceptable treatment for TCGA tumor'.
items[].classification	string	Annotation classification, .e.g 'CenterNotification', 'Notification', 'Observation', or 'Redaction'.
items[].endpoint_type	string	Which type of GDC Annotation API was used, either legacy or current
items[].entity_barcode	string	The TCGA barcode that the annottion is associated with
items[].entity_gdc_id	string	Id assigned by the GDC to the entity
items[].entity_type	string	Entity type, e.g. 'Case', 'Aliquot', 'Analyte', 'Portion'', 'Slide', or 'Sample'.
items[].notes	string	Notes on the annotation
items[].program_name	string	The program name, e.g. 'TCGA' (the only program with annotations)
items[].project_short_name	string	The project id, e.g. 'TCGA-BRCA', 'TCGA-OV'.
items[].sample_barcode	string	Sample barcode.
items[].status	string	Status of the annotation, e.g. 'Approved', 'Rescinded'

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

TARGET Endpoints

cohorts().preview()

Takes a JSON object of filters in the request body and returns a "preview" of the cohort that would result from passing a similar request to the cohort **save** endpoint. This preview consists of two lists: the lists of case barcodes, and the list of sample barcodes. Authentication is not required.

Example:

```
curl "https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/
↪cohorts/preview?program_short_name=TARGET-ALL-P2&program_short_name=TARGET-
↪RT&age_at_diagnosis_lte=20"
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2


def get_unauthorized_service():
        api = 'isb_cgc_target_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
→api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
→http=httplib2.Http())


service = get_unauthorized_service()
body = {'program_short_name': ['TARGET-ALL-P2', 'TARGET-RT'], 'age_at_
→diagnosis_gte': 90}
data = service.cohorts().preview(**body).execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/tcga/
→cohorts/preview?fields=case_count%2Ccases%2Csample_count%2Csamples
```

{

    "Common": {

        "project_short_name": ["TARGET-ALL-P2", "TARGET-RT"

        ]

    }, "Clinical": {

        "age_at_diagnosis_lte": 20

    }

}

Parameters

Parameter name	Value	Description
age_at_diagnosis	integer	Optional.
age_at_diagnosis_gte	integer	Optional.
age_at_diagnosis_lte	integer	Optional.
case_barcode	string	Optional.
case_gdc_id	string	Optional.
days_to_birth	integer	Optional.
days_to_birth_gte	integer	Optional.
days_to_birth_lte	integer	Optional.
days_to_death	integer	Optional.
days_to_death_gte	integer	Optional.
days_to_death_lte	integer	Optional.
days_to_last_followup	integer	Optional.
days_to_last_followup_gte	integer	Optional.
days_to_last_followup_lte	integer	Optional.
days_to_last_known_alive	integer	Optional.

Continued on next page

Table 7 – continued from previous page

Parameter name	Value	Description
days_to_last_known_alive_gte	integer	Optional.
days_to_last_known_alive_lte	integer	Optional.
disease_code	string	Optional.
endpoint_type	string	Optional.
ethnicity	string	Optional.
event_free_survival	integer	Optional.
event_free_survival_gte	integer	Optional.
event_free_survival_lte	integer	Optional.
first_event	string	Optional.
gender	string	Optional.
program_name	string	Optional.
project_short_name	string	Optional.
protocol	string	Optional.
race	string	Optional.
sample_barcode	string	Optional.
sample_gdc_id	string	Optional.
sample_type	string	Optional.
summary_file_count	integer	Optional.
summary_file_count_gte	integer	Optional.
summary_file_count_lte	integer	Optional.
tumor_code	string	Optional.
vital_status	string	Optional.
wbc_at_diagnosis	number	Optional.
wbc_at_diagnosis_gte	number	Optional.
wbc_at_diagnosis_lte	number	Optional.
year_of_diagnosis	integer	Optional.
year_of_diagnosis_gte	integer	Optional.
year_of_diagnosis_lte	integer	Optional.
year_of_last_follow_up	integer	Optional.
year_of_last_follow_up_gte	integer	Optional.
year_of_last_follow_up_lte	integer	Optional.

Response

If successful, this method returns a response body with the following structure:

```
{
  "case_count": integer,
  "cases": [string],
  "sample_count": integer,
  "samples": [string]
}
```

Parameter name	Value	Description
case_count	integer	Number of cases in the cohort.
cases[]	list	List of cases barcodes in the cohort.
sample_count	integer	Number of samples in the cohort.
samples[]	list	List of sample barcodes in the cohort.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cohorts().create()

Creates and saves a cohort. Takes a JSON object in the request body to use as the cohort's filters. Authentication is required. Returns information about the saved cohort, including the number of cases and the number of samples in that cohort.

Example:

```
python isb_curl.py "https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_
→target_api/v3/cohorts/create?name={COHORT NAME}" -H "Content-Type:␣
→application/json" -d '{"program_short_name": ["TARGET-ALL-P2", "TARGET-RT
→"], "age_at_diagnosis_lte": 60}'
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
from oauth2client.client import OAuth2WebServerFlow
from oauth2client import tools
from oauth2client.file import Storage
import httplib2
import os

CLIENT_ID = '907668440978-0ol0griu70qkeb6k3gnn2vipfa5mgl60.apps.
→googleusercontent.com'
CLIENT_SECRET = 'To_WJH7-1V-TofhNGcEqmEYi'
EMAIL_SCOPE = 'https://www.googleapis.com/auth/userinfo.email'
DEFAULT_STORAGE_FILE = os.path.join(os.path.expanduser('~'), '.isb_
→credentials')


def get_credentials():
        oauth_flow_args = ['--noauth_local_webserver']
        storage = Storage(DEFAULT_STORAGE_FILE)
        credentials = storage.get()
        if not credentials or credentials.invalid:
                flow = OAuth2WebServerFlow(CLIENT_ID, CLIENT_SECRET, EMAIL_
→SCOPE)
                flow.auth_uri = flow.auth_uri.rstrip('/') + '?approval_
→prompt=force'
                credentials = tools.run_flow(flow, storage, tools.argparser.
→parse_args(oauth_flow_args))
        return credentials


def get_authorized_service():
        api = 'isb_cgc_target_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,␣
→api, version)
        credentials = get_credentials()
        http = credentials.authorize(httplib2.Http())
        if credentials.access_token_expired or credentials.invalid:
                credentials.refresh(http)
        authorized_service = build(api, version,␣
→discoveryServiceUrl=discovery_url, http=http)
```

(continues on next page)

```
        return authorized_service

service = get_authorized_service()
body = {'program_short_name': ['TARGET-ALL-P2', 'TARGET-RT'], 'age_at_
→diagnosis_gte': 90}
data = service.cohorts().create(name=name, body=body).execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/
→target/cohorts/create?name=Cohort_test&fields=case_count%2Cfilters%2Cid
→%2Clast_date_saved%2Cname%2Csample_count
```

{

> **"Common": {**
>
> > **"project_short_name": [** "TARGET-ALL-P2", "TARGET-RT"
> >
> >]
>
> }, "Clinical": {
>
> > "age_at_diagnosis_lte": 60
>
> }

}

Parameters

Parameter name	Value	Description
name	string	Required.

Request body

In the request body, supply a metadata resource with the following properties:

```
{
  "age_at_diagnosis": [integer],
  "age_at_diagnosis_gte": integer,
  "age_at_diagnosis_lte": integer,
  "case_barcode": [string],
  "case_gdc_id": [string],
  "days_to_birth": [integer],
  "days_to_birth_gte": integer,
  "days_to_birth_lte": integer,
  "days_to_death": [integer],
  "days_to_death_gte": integer,
  "days_to_death_lte": integer,
  "days_to_last_followup": [integer],
  "days_to_last_followup_gte": integer,
  "days_to_last_followup_lte": integer,
  "days_to_last_known_alive": [integer],
  "days_to_last_known_alive_gte": integer,
  "days_to_last_known_alive_lte": integer,
  "disease_code": [string],
```

```
    "endpoint_type": [string],
    "ethnicity": [string],
    "event_free_survival": [integer],
    "event_free_survival_gte": integer,
    "event_free_survival_lte": integer,
    "first_event": [string],
    "gender": [string],
    "program_name": [string],
    "project_short_name": [string],
    "protocol": [string],
    "race": [string],
    "sample_barcode": [string],
    "sample_gdc_id": [string],
    "sample_type": [string],
    "summary_file_count": [integer],
    "summary_file_count_gte": integer,
    "summary_file_count_lte": integer,
    "tumor_code": [string],
    "vital_status": [string],
    "wbc_at_diagnosis": [number],
    "wbc_at_diagnosis_gte": number,
    "wbc_at_diagnosis_lte": number,
    "year_of_diagnosis": [integer],
    "year_of_diagnosis_gte": integer,
    "year_of_diagnosis_lte": integer,
    "year_of_last_follow_up": [integer],
    "year_of_last_follow_up_gte": integer,
    "year_of_last_follow_up_lte": integer
}
```

Parameter name	Value	Description
age_at_diagnosis[]	list	Optional.
age_at_diagnosis_gte	integer	Optional.
age_at_diagnosis_lte	integer	Optional.
case_barcode[]	list	Optional.
case_gdc_id[]	list	Optional.
days_to_birth[]	list	Optional.
days_to_birth_gte	integer	Optional.
days_to_birth_lte	integer	Optional.
days_to_death[]	list	Optional.
days_to_death_gte	integer	Optional.
days_to_death_lte	integer	Optional.
days_to_last_followup[]	list	Optional.
days_to_last_followup_gte	integer	Optional.
days_to_last_followup_lte	integer	Optional.
days_to_last_known_alive[]	list	Optional.
days_to_last_known_alive_gte	integer	Optional.
days_to_last_known_alive_lte	integer	Optional.
disease_code[]	list	Optional. Possible values include: 'ALL', 'AML', 'CCSK', 'NBL', 'OS', 'RT', 'WT'.
endpoint_type[]	list	Optional. Possible values include: 'current', 'legacy'.

Continued on next page

Table 8 – continued from previous page

Parameter name	Value	Description
ethnicity[]	list	Optional. Possible values include: 'Hispanic or Latino', 'Not Hispanic or Latino'.
event_free_survival[]	list	Optional.
event_free_survival_gte	integer	Optional.
event_free_survival_lte	integer	Optional.
first_event[]	list	Optional. Possible values include: 'Censored', 'Death', 'Death without remission', 'Event', 'Induction failure', 'Progression', 'Relapse', 'Second Malignant Neoplasm'.
gender[]	list	Optional. Possible values include: 'Female', 'Male'.
program_name[]	list	Optional. Possible values include: 'TARGET'.
project_short_name[]	list	Optional. Possible values include: 'TARGET-ALL-P1', 'TARGET-ALL-P2', 'TARGET-AML', 'TARGET-CCSK', 'TARGET-NBL', 'TARGET-OS', 'TARGET-RT', 'TARGET-WT'.
protocol[]	list	Optional.
race[]	list	Optional. Possible values include: 'American Indian or Alaska Native', 'Asian', 'Black or African American', 'Native Hawaiian or other Pacific Islander', 'Other', 'White'.
sample_barcode[]	list	Optional.
sample_gdc_id[]	list	Optional.
sample_type[]	list	Optional. Possible values include: '01', '02', '03', '04', '06', '08', '09', '10', '11', '13', '14', '15', '20', '40', '41', '42', '50', '60'.
summary_file_count[]	list	Optional.
summary_file_count_gte	integer	Optional.
summary_file_count_lte	integer	Optional.
tumor_code[]	list	Optional. Possible values include: '00', '10', '20', '21', '30', '40', '50', '51', '52'.
vital_status[]	list	Optional. Possible values include: 'alive', 'dead'.
wbc_at_diagnosis[]	list	Optional.
wbc_at_diagnosis_gte	number	Optional.
wbc_at_diagnosis_lte	number	Optional.
year_of_diagnosis[]	list	Optional.
year_of_diagnosis_gte	integer	Optional.
year_of_diagnosis_lte	integer	Optional.
year_of_last_follow_up[]	list	Optional.
year_of_last_follow_up_gte	integer	Optional.
year_of_last_follow_up_lte	integer	Optional.

Response

If successful, this method returns a response body with the following structure:

```
{
  "case_count": integer,
```

```
  "filters": [
    {
      "name": string,
      "value": string
    }
  ],
  "id": string,
  "last_date_saved": string,
  "name": string,
  "sample_count": integer
}
```

Parameter name	Value	Description
case_count	integer	Number of unique case barcodes in the cohort.
filters[]	list	List of filters applied to create cohort, if any.
filters[].name	string	Names of filtering parameters used to create the cohort.
filters[].value	string	Values of filtering parameters used to create the cohort.
id	string	Cohort id.
last_date_saved	string	Last date the cohort was saved.
name	string	Name of cohort.
sample_count	integer	Number of unique sample barcodes in the cohort.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cases().get()

Returns information about a specific case, including a list of samples and aliquots derived from this case. Takes a case barcode (of length 16, *eg* TARGET-51-PALFYG) as a required parameter. User does not need to be authenticated.

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/
→target/cases/TARGET-10-DCC001
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_target_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
```

```
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
→api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
→http=httplib2.Http())

service = get_unauthorized_service()
data = service.cases().get(case_barcode='TARGET-10-DCC001').execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/target/
→cases/{case_barcode}
```

Parameters

Parameter name	Value	Description
case_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "aliquots": [string],
  "clinical_data": {
    "age_at_diagnosis": integer,
    "case_barcode": string,
    "case_gdc_id": string,
    "days_to_birth": integer,
    "days_to_death": integer,
    "days_to_last_followup": integer,
    "days_to_last_known_alive": integer,
    "disease_code": string,
    "endpoint_type": string,
    "ethnicity": string,
    "event_free_survival": integer,
    "first_event": string,
    "gender": string,
    "program_name": string,
    "project_short_name": string,
    "protocol": string,
    "race": string,
    "sample_barcode": string,
    "sample_gdc_id": string,
    "sample_type": string,
    "summary_file_count": integer,
    "tumor_code": string,
    "vital_status": string,
    "wbc_at_diagnosis": number,
    "year_of_diagnosis": integer,
    "year_of_last_follow_up": integer
  },
  "samples": [string]
}
```

Parameter name	Value	Description
aliquots[]	list	List of barcodes of aliquots taken from this patient.
clinical_data	nested object	The clinical data about the patient.
clinical_data.age_at_diagnosis	integer	Age at which a condition or disease was first diagnosed in years.
clinical_data.case_barcode	string	Case barcode.
clinical_data.case_gdc_id	string	The GDC assigned id for the case
clinical_data.days_to_birth	integer	Time interval from a person's date of birth to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_death	integer	Time interval from a person's date of death to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_last_followup	integer	Time interval from the date of last followup to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_last_known_alive	integer	The number of days between diagnosis and when the individual was last known to be alive.
clinical_data.disease_code	string	The short name for the type of disease
clinical_data.endpoint_type	string	Which type of GDC Case API was used, either legacy or current
clinical_data.ethnicity	string	The text for reporting information about ethnicity based on the Office of Management and Budget (OMB) categories.
clinical_data.event_free_survival	integer	The length of time after primary treatment for a cancer ends that the patient remains free of certain complications or events.
clinical_data.first_event	string	The first event after the diagnosis of cancer.
clinical_data.gender	string	Text designations that identify gender.
clinical_data.program_name	string	Project name, e.g. 'TCGA'.
clinical_data.project_short_name	string	Tumor type abbreviation, e.g. 'BRCA'.
clinical_data.protocol	string	A list detailed plans of scientific or medical experiments, treatments, or procedures.
clinical_data.race	string	The text for reporting information about race based on the Office of Management and Budget (OMB) categories.
clinical_data.sample_barcode	string	The barcode assigned by TCGA to a sample from a Participant.
clinical_data.sample_gdc_id	string	The GDC assigned id for the sample
clinical_data.sample_type	string	The type of the sample tumor or normal tissue cell or blood sample provided by a participant.
clinical_data.summary_file_count	integer	The count of files associated with the sample
clinical_data.tumor_code	string	Code representing the type of tumor.
clinical_data.vital_status	string	The survival state of the person registered on the protocol.
clinical_data.wbc_at_diagnosis	number	White blood cell range at diagnosis
clinical_data.year_of_diagnosis	integer	Numeric value to represent the year of an individual's initial pathologic diagnosis of cancer.
clinical_data.year_of_last_follow_up	integer	Numeric value to represent the year of an individual's last follow up.
samples[]	list	List of barcodes of samples taken from this patient.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cases().get_list()

Given a list of case barcodes (of length 11, *eg* TARGET-51-PALFYG), this endpoint returns all available "biospecimen" information about them, including a list of samples and aliquots derived from them. Takes a list of case barcodes (of length 12, *eg* TARGET-51-PALFYG) as a required data payload. User does not need to be authenticated.

Example:

```
curl --data '{"case_barcodes": ["TARGET-10-DCC001","TARGET-52-PASGGN"]}'␣
↪https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/target/
↪cases
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_target_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,␣
↪api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,␣
↪http=httplib2.Http())

service = get_unauthorized_service()
data = service.cases().get(body={"case_barcodes": ["TARGET-10-DCC001",
↪"TARGET-52-PASGGN"]}).execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/
↪target/cases
```

{

> **"case_barcodes": [** "TARGET-10-DCC001", "TARGET-52-PASGGN"
>
>]

}

Parameters

Parameter name	Value	Description
widths 50	10	50
case_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "aliquots": [string],
  "clinical_data": {
    "age_at_diagnosis": integer,
    "case_barcode": string,
    "case_gdc_id": string,
    "days_to_birth": integer,
    "days_to_death": integer,
    "days_to_last_followup": integer,
    "days_to_last_known_alive": integer,
    "disease_code": string,
    "endpoint_type": string,
    "ethnicity": string,
    "event_free_survival": integer,
    "first_event": string,
    "gender": string,
    "program_name": string,
    "project_short_name": string,
    "protocol": string,
    "race": string,
    "sample_barcode": string,
    "sample_gdc_id": string,
    "sample_type": string,
    "summary_file_count": integer,
    "tumor_code": string,
    "vital_status": string,
    "wbc_at_diagnosis": number,
    "year_of_diagnosis": integer,
    "year_of_last_follow_up": integer
  },
  "samples": [string],
  "case_barcode": string
}
```

Parameter name	Value	Description
aliquots[]	list	List of barcodes of aliquots taken from this patient.
clinical_data	nested object	The clinical data about the patient.
clinical_data.age_at_diagnosis	integer	Age at which a condition or disease was first diagnosed in years.
clinical_data.case_barcode	string	Case barcode.
clinical_data.case_gdc_id	string	The GDC assigned id for the case
clinical_data.days_to_birth	integer	Time interval from a person's date of birth to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_death	integer	Time interval from a person's date of death to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_last_followup	integer	Time interval from the date of last followup to the date of initial pathologic diagnosis, represented as a calculated number of days.
clinical_data.days_to_last_known_alive	integer	The number of days between diagnosis and when the individual was last known to be alive.
clinical_data.disease_code	string	The short name for the type of disease
clinical_data.endpoint_type	string	Which type of GDC Case API was used, either legacy or current
clinical_data.ethnicity	string	The text for reporting information about ethnicity based on the Office of Management and Budget (OMB) categories.
clinical_data.event_free_survival	integer	The length of time after primary treatment for a cancer ends that the patient remains free of certain complications or events.
clinical_data.first_event	string	The first event after the diagnosis of cancer.
clinical_data.gender	string	Text designations that identify gender.
clinical_data.program_name	string	Project name, e.g. 'TCGA'.
clinical_data.project_short_name	string	Tumor type abbreviation, e.g. 'BRCA'.
clinical_data.protocol	string	A list detailed plans of scientific or medical experiments, treatments, or procedures.
clinical_data.race	string	The text for reporting information about race based on the Office of Management and Budget (OMB) categories.
clinical_data.sample_barcode	string	The barcode assigned by TCGA to a sample from a Participant.
clinical_data.sample_gdc_id	string	The GDC assigned id for the sample
clinical_data.sample_type	string	The type of the sample tumor or normal tissue cell or blood sample provided by a participant.
clinical_data.summary_file_count	integer	The count of files associated with the sample
clinical_data.tumor_code	string	Code representing the type of tumor.
clinical_data.vital_status	string	The survival state of the person registered on the protocol.
clinical_data.wbc_at_diagnosis	number	White blood cell range at diagnosis
clinical_data.year_of_diagnosis	integer	Numeric value to represent the year of an individual's initial pathologic diagnosis of cancer.
clinical_data.year_of_last_follow_up	integer	Numeric value to represent the year of an individual's last follow up.
samples[]	list	List of barcodes of samples taken from this patient.

1.6. Programmatic Access

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

samples().cloud_storage_file_paths()

Takes a sample barcode as a required parameter and returns cloud storage paths to files associated with that sample.

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/
↪target/samples/TARGET-10-DCC001-03A/cloud_storage_file_paths
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_target_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
↪http=httplib2.Http())

service = get_unauthorized_service()
data = service.samples().cloud_storage_file_paths(sample_barcode='TARGET-10-
↪DCC001-03A').execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/target/
↪samples/{sample_barcode}/cloud_storage_file_paths
```

Parameters

Parameter name	Value	Description
analysis_workflow_type	string	Optional.
data_category	string	Optional.
data_format	string	Optional.
data_type	string	Optional.
experimental_strategy	string	Optional.
genomic_build	string	Optional.
platform	string	Optional.
sample_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "cloud_storage_file_paths": [string],
  "count": integer
}
```

Parameter name	Value	Description
cloud_storage_file_paths[]	list	List of Google Cloud Storage paths of files associated with the cohort.
count	integer	Number of Google Cloud Storage paths returned for the cohort.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

samples().get()

Given a sample barcode (of length 20-22, *eg* TARGET-51-PALFYG-01A), this endpoint returns all available "biospecimen" information about this sample, the associated case barcode, a list of associated aliquots, and a list of "data_details" blocks describing each of the data files associated with this sample

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/
→target/samples/TARGET-10-DCC001-03A
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_target_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
→api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
→http=httplib2.Http())

service = get_unauthorized_service()
data = service.samples().get(sample_barcode='TARGET-10-DCC001-03A').execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/target/
→samples/{sample_barcode}
```

Parameters

Parameter name	Value	Description
analysis_workflow_type	string	Optional.
data_category	string	Optional.
data_format	string	Optional.
data_type	string	Optional.
endpoint_type	string	Optional.
experimental_strategy	string	Optional.
platform	string	Optional.
sample_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "aliquots": [string],
  "biospecimen_data": {
    "age_at_diagnosis": integer,
    "case_barcode": string,
    "case_gdc_id": string,
    "days_to_birth": integer,
    "days_to_death": integer,
    "days_to_last_followup": integer,
    "days_to_last_known_alive": integer,
    "disease_code": string,
    "endpoint_type": string,
    "ethnicity": string,
    "event_free_survival": integer,
    "first_event": string,
    "gender": string,
    "program_name": string,
    "project_short_name": string,
    "protocol": string,
    "race": string,
    "sample_barcode": string,
    "sample_gdc_id": string,
    "sample_type": string,
    "summary_file_count": integer,
    "tumor_code": string,
    "vital_status": string,
    "wbc_at_diagnosis": number,
    "year_of_diagnosis": integer,
    "year_of_last_follow_up": integer
  },
  "case": string,
  "data_details": [
    {
      "access": string,
      "analysis_workflow_type": string,
      "data_category": string,
      "data_format": string,
      "data_type": string,
      "disease_code": string,
      "endpoint_type": string,
      "experimental_strategy": string,
```

```
        "file_gdc_id": string,
        "file_name": string,
        "file_name_key": string,
        "file_size": string,
        "index_file_name": string,
        "platform": string,
        "program_name": string,
        "project_short_name": string,
        "sample_barcode": string,
        "sample_gdc_id": string,
        "sample_type": string
    }
  ],
  "data_details_count": integer
}
```

Parameter name	Value	Description
aliquots[]	list	List of barcodes of aliquots taken from this participant.
biospecimen_data	nested object	Biospecimen data about the sample.
biospecimen_data.age_at_diagnosis	integer	Age at which a condition or disease was first diagnosed in years.
biospecimen_data.case_barcode	string	Case barcode.
biospecimen_data.case_gdc_id	string	The GDC assigned id for the case
biospecimen_data.days_to_birth	integer	Time interval from a person's date of birth to the date of initial pathologic diagnosis, represented as a calculated number of days.
biospecimen_data.days_to_death	integer	Time interval from a person's date of death to the date of initial pathologic diagnosis, represented as a calculated number of days.
biospecimen_data.days_to_last_followup	integer	Time interval from the date of last followup to the date of initial pathologic diagnosis, represented as a calculated number of days.
biospecimen_data.days_to_last_known_alive	integer	The number of days between diagnosis and when the individual was last known to be alive.
biospecimen_data.disease_code	string	The short name for the type of disease
biospecimen_data.endpoint_type	string	Which type of GDC Case API was used, either legacy or current
biospecimen_data.ethnicity	string	The text for reporting information about ethnicity based on the Office of Management and Budget (OMB) categories.
biospecimen_data.event_free_survival	integer	The length of time after primary treatment for a cancer ends that the patient remains free of certain complications or events.
biospecimen_data.first_event	string	The first event after the diagnosis of cancer.
biospecimen_data.gender	string	Text designations that identify gender.
biospecimen_data.program_name	string	Project name, e.g. 'TCGA'.
biospecimen_data.project_short_name	string	Tumor type abbreviation, e.g. 'BRCA'.

Table 9 – continued from previous page

Parameter name	Value	Description
biospecimen_data.protocol	string	A list detailed plans of scientific or medical experiments, treatments, or procedures.
biospecimen_data.race	string	The text for reporting information about race based on the Office of Management and Budget (OMB) categories.
biospecimen_data.sample_barcode	string	The barcode assigned by TCGA to a sample from a Participant.
biospecimen_data.sample_gdc_id	string	The GDC assigned id for the sample
biospecimen_data.sample_type	string	The type of the sample tumor or normal tissue cell or blood sample provided by a participant.
biospecimen_data.summary_file_count	integer	The count of files associated with the sample
biospecimen_data.tumor_code	string	Code representing the type of tumor.
biospecimen_data.vital_status	string	The survival state of the person registered on the protocol.
biospecimen_data.wbc_at_diagnosis	number	White blood cell range at diagnosis
biospecimen_data.year_of_diagnosis	integer	Numeric value to represent the year of an individual's initial pathologic diagnosis of cancer.
biospecimen_data.year_of_last_follow_up	integer	Numeric value to represent the year of an individual's last follow up.
case	string	Case barcode.
data_details[]	list	List of information about each file associated with the sample barcode.
data_details[].access	string	An indication of the security protocol necessary to fulfill in order to access the data from the file, e.g. open, controlled.
data_details[].analysis_workflow_type	string	The type of workflow used to generate the data file, e.g. 'BWA-aln', 'STAR 2-Pass', 'BWA with Mark Duplicates and Cocleaning'
data_details[].data_category	string	The higher level categorization of the data_type in the file, e.g. 'Biospecimen', 'Clinical', 'Raw sequencing data', 'Simple nucleotide variation'
data_details[].data_format	string	The format of the data file, e.g. 'BAM', 'BCR XML', 'TXT'
data_details[].data_type	string	Data type stored in Google Cloud Storage, e.g. 'Clinical Supplement', 'Biospecimen Supplement', 'Aligned reads', 'Genotypes', 'Diagnostic image'
data_details[].disease_code	string	The disease abbeviation, e.g. 'ACC', 'UVM', 'ALL', 'WT'
data_details[].endpoint_type	string	The GDC files API the data file information was gottern from, e.g. 'legacy', 'current'
data_details[].experimental_strategy	string	The sequencing, array or other strategy used to generate the data file, e.g. 'RNA-Seq', 'WGS', 'Genotyping array'
data_details[].file_gdc_id	string	The GDC assigned id for the file

Continued on next page

Table 9 – continued from previous page

Parameter name	Value	Description
data_details[].file_name	string	Name of the datafile stored on the GDC system.
data_details[].file_name_key	string	Google Cloud Storage path to file.
data_details[].file_size	string	The size of the file
data_details[].index_file_name	string	For BAM files, the name of its index file
data_details[].platform	string	The sequencing or array platform used, e.g. Illumina HiSeq, Ion Torrent PGM, Affymetrix SNP Array 6.0.
data_details[].program_name	string	The program for which the data was generated, e.g. 'CCLE', 'TARGET','TCGA'.
data_details[].project_short_name	string	The id of the project, e.g. 'CCLE-ACC', 'CCLE-UVM', 'TARGET-ALL-P1', ' TARGET-WT', 'TCGA-ACC', 'TCGA-UVM'
data_details[].sample_barcode	string	Sample barcode.
data_details[].sample_gdc_id	string	The GDC assigned id for the sample
data_details[].sample_type	string	The sample type, e.g. '01', '10', '11'
data_details_count	integer	Number of files associated with the sample barcode.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

samples().get_list()

Given a list of sample barcodes (of length 16, *eg* TARGET-B9-7268-01A), this endpoint returns all available "biospecimen" information about this sample, the associated case barcode, a list of associated aliquots, and a list of "data_details" blocks describing each of the data files associated with this sample.

Example:

```
curl --data '{"sample_barcodes": ["TARGET-10-DCC001-03A","TARGET-52-PASGGN-
↪01A"]}' https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/
↪target/samples
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_target_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
↪http=httplib2.Http())
```

(continues on next page)

```
service = get_unauthorized_service()
data = service.samples().get(body={"case_barcodes": ["TARGET-52-PASGGN-01A",
↪"TARGET-10-DCC001-03A"]}).execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/
↪target/samples
```

{

"sample_barcodes": ["TARGET-10-DCC001-03A", "TARGET-52-PASGGN-01A"

]

}

Parameters

Parameter name	Value	Description
widths 50	10	50
analysis_workflow_type	string	Optional.
data_category	string	Optional.
data_format	string	Optional.
data_type	string	Optional.
endpoint_type	string	Optional.
experimental_strategy	string	Optional.
platform	string	Optional.
sample_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "aliquots": [string],
  "biospecimen_data": {
    "age_at_diagnosis": integer,
    "case_barcode": string,
    "case_gdc_id": string,
    "days_to_birth": integer,
    "days_to_death": integer,
    "days_to_last_followup": integer,
    "days_to_last_known_alive": integer,
    "disease_code": string,
    "endpoint_type": string,
    "ethnicity": string,
    "event_free_survival": integer,
    "first_event": string,
    "gender": string,
    "program_name": string,
    "project_short_name": string,
```

```
    "protocol": string,
    "race": string,
    "sample_barcode": string,
    "sample_gdc_id": string,
    "sample_type": string,
    "summary_file_count": integer,
    "tumor_code": string,
    "vital_status": string,
    "wbc_at_diagnosis": number,
    "year_of_diagnosis": integer,
    "year_of_last_follow_up": integer
  },
  "case_barcode": string,
  "data_details": [
    {
      "access": string,
      "analysis_workflow_type": string,
      "data_category": string,
      "data_format": string,
      "data_type": string,
      "disease_code": string,
      "endpoint_type": string,
      "experimental_strategy": string,
      "file_gdc_id": string,
      "file_name": string,
      "file_name_key": string,
      "file_size": string,
      "index_file_name": string,
      "platform": string,
      "program_name": string,
      "project_short_name": string,
      "sample_barcode": string,
      "sample_gdc_id": string,
      "sample_type": string
    }
  ],
  "data_details_count": integer,
  "sample_barcode": string
}
```

Parameter name	Value	Description
widths 50	10	50
aliquots[]	list	List of barcodes of aliquots taken from this participant.
biospecimen_data	nested object	Biospecimen data about the sample.
biospecimen_data.age_at_diagnosis	integer	Age at which a condition or disease was first diagnosed in years.
biospecimen_data.case_barcode	string	Case barcode.
biospecimen_data.case_gdc_id	string	The GDC assigned id for the case

Continued on next page

Table 10 – continued from previous page

Parameter name	Value	Description
widths 50	10	50
biospecimen_data.days_to_birth	integer	Time interval from a person's date of birth to the date of initial pathologic diagnosis, represented as a calculated number of days.
biospecimen_data.days_to_death	integer	Time interval from a person's date of death to the date of initial pathologic diagnosis, represented as a calculated number of days.
biospecimen_data.days_to_last_followup	integer	Time interval from the date of last followup to the date of initial pathologic diagnosis, represented as a calculated number of days.
biospecimen_data.days_to_last_known_alive	integer	The number of days between diagnosis and when the individual was last known to be alive.
biospecimen_data.disease_code	string	The short name for the type of disease
biospecimen_data.endpoint_type	string	Which type of GDC Case API was used, either legacy or current
biospecimen_data.ethnicity	string	The text for reporting information about ethnicity based on the Office of Management and Budget (OMB) categories.
biospecimen_data.event_free_survival	integer	The length of time after primary treatment for a cancer ends that the patient remains free of certain complications or events.
biospecimen_data.first_event	string	The first event after the diagnosis of cancer.
biospecimen_data.gender	string	Text designations that identify gender.
biospecimen_data.program_name	string	Project name, e.g. 'TCGA'.
biospecimen_data.project_short_name	string	Tumor type abbreviation, e.g. 'BRCA'.
biospecimen_data.protocol	string	A list detailed plans of scientific or medical experiments, treatments, or procedures.
biospecimen_data.race	string	The text for reporting information about race based on the Office of Management and Budget (OMB) categories.
biospecimen_data.sample_barcode	string	The barcode assigned by TCGA to a sample from a Participant.

Continued on next page

Table 10 – continued from previous page

Parameter name	Value	Description
widths 50	10	50
biospecimen_data.sample_gdc_id	string	The GDC assigned id for the sample
biospecimen_data.sample_type	string	The type of the sample tumor or normal tissue cell or blood sample provided by a participant.
biospecimen_data.summary_file_count	integer	The count of files associated with the sample
biospecimen_data.tumor_code	string	Code representing the type of tumor.
biospecimen_data.vital_status	string	The survival state of the person registered on the protocol.
biospecimen_data.wbc_at_diagnosis	number	White blood cell range at diagnosis
biospecimen_data.year_of_diagnosis	integer	Numeric value to represent the year of an individual's initial pathologic diagnosis of cancer.
biospecimen_data.year_of_last_follow_up	integer	Numeric value to represent the year of an individual's last follow up.
case	string	Case barcode.
data_details[]	list	List of information about each file associated with the sample barcode.
data_details[].access	string	An indication of the security protocol necessary to fulfill in order to access the data from the file, e.g. open, controlled.
data_details[].analysis_workflow_type	string	The type of workflow used to generate the data file, e.g. 'BWA-aln', 'STAR 2-Pass', 'BWA with Mark Duplicates and Cocleaning'
data_details[].data_category	string	The higher level categorization of the data_type in the file, e.g. 'Biospecimen', 'Clinical', 'Raw sequencing data', 'Simple nucleotide variation'
data_details[].data_format	string	The format of the data file, e.g. 'BAM', 'BCR XML', 'TXT'
data_details[].data_type	string	Data type stored in Google Cloud Storage, e.g. 'Clinical Supplement', 'Biospecimen Supplement', 'Aligned reads', 'Genotypes', 'Diagnostic image'
data_details[].disease_code	string	The disease abbeviation, e.g. 'ACC', 'UVM', 'ALL', 'WT'

Continued on next page

Table 10 – continued from previous page

Parameter name	Value	Description
widths 50	10	50
data_details[].endpoint_type	string	The GDC files API the data file information was gottern from, e.g. 'legacy', 'current'
data_details[].experimental_strategy	string	The sequencing, array or other strategy used to generate the data file, e.g. 'RNA-Seq', 'WGS', 'Genotyping array'
data_details[].file_gdc_id	string	The GDC assigned id for the file
data_details[].file_name	string	Name of the datafile stored on the GDC system.
data_details[].file_name_key	string	Google Cloud Storage path to file.
data_details[].file_size	string	The size of the file
data_details[].index_file_name	string	For BAM files, the name of its index file
data_details[].platform	string	The sequencing or array platform used, e.g. Illumina HiSeq, Ion Torrent PGM, Affymetrix SNP Array 6.0.
data_details[].program_name	string	The program for which the data was generated, e.g. 'CCLE', 'TARGET','TCGA'.
data_details[].project_short_name	string	The id of the project, e.g. 'CCLE-ACC', 'CCLE-UVM', 'TARGET-ALL-P1', ' TARGET-WT', 'TCGA-ACC', 'TCGA-UVM'
data_details[].sample_barcode	string	Sample barcode.
data_details[].sample_gdc_id	string	The GDC assigned id for the sample
data_details[].sample_type	string	The sample type, e.g. '01', '10', '11'
data_details_count	integer	Number of files associated with the sample barcode.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

users().get()

Returns the dbGaP authorization status of the user.

Example:

```
python isb_curl.py https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_
↪target_api/v3/target/users
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
from oauth2client.client import OAuth2WebServerFlow
from oauth2client import tools
from oauth2client.file import Storage
import httplib2
import os

CLIENT_ID = '907668440978-0ol0griu70qkeb6k3gnn2vipfa5mgl60.apps.
↪googleusercontent.com'
CLIENT_SECRET = 'To_WJH7-1V-TofhNGcEqmEYi'
EMAIL_SCOPE = 'https://www.googleapis.com/auth/userinfo.email'
DEFAULT_STORAGE_FILE = os.path.join(os.path.expanduser('~'), '.isb_
↪credentials')


def get_credentials():
        oauth_flow_args = ['--noauth_local_webserver']
        storage = Storage(DEFAULT_STORAGE_FILE)
        credentials = storage.get()
        if not credentials or credentials.invalid:
                flow = OAuth2WebServerFlow(CLIENT_ID, CLIENT_SECRET, EMAIL_
↪SCOPE)
                flow.auth_uri = flow.auth_uri.rstrip('/') + '?approval_
↪prompt=force'
                credentials = tools.run_flow(flow, storage, tools.argparser.
↪parse_args(oauth_flow_args))
        return credentials


def get_authorized_service():
        api = 'isb_cgc_target_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        credentials = get_credentials()
        http = credentials.authorize(httplib2.Http())
        if credentials.access_token_expired or credentials.invalid:
                credentials.refresh(http)
        authorized_service = build(api, version,
↪discoveryServiceUrl=discovery_url, http=http)
        return authorized_service


service = get_authorized_service()
data = service.users().get().execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_target_api/v3/target/
↪users
```

Parameters

None

Response

If successful, this method returns a response body with the following structure:

```
{
  "dbGaP_allowed": boolean,
  "dbGaP_authorized": boolean,
  "message": string
}
```

Parameter name	Value	Description
dbGaP_authorized	boolean	True or false.
message	string	Message indicating the authorization status of the user.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

CCLE Endpoints

cohorts().preview()

Takes a JSON object of filters in the request body and returns a "preview" of the cohort that would result from passing a similar request to the cohort **save** endpoint. This preview consists of two lists: the lists of case barcodes, and the list of sample barcodes. Authentication is not required.

Example:

```
curl "https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/
↪cohorts/preview?program_short_name=CCLE-BLCA&program_short_name=CCLE-LUSC&
↪gender=Male"
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_ccle_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
↪http=httplib2.Http())

service = get_unauthorized_service()
body = {'program_short_name': ['CCLE-BLCA', 'CCLE-LUSC'], 'gender': Male}
data = service.cohorts().preview(**body).execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/tcga/
↪cohorts/preview?fields=case_count%2Ccases%2Csample_count%2Csamples
```

{

> **"Common": {**
>
> > **"project_short_name": [** "CCLE-BLCA", "CCLE-LUSC"
> >
> >]
>
> **}, "Clinical": {**
>
> > **"gender": [** "Male"
> >
> >]
>
> **}**

}

Parameters

Parameter name	Value	Description
case_barcode	string	Optional.
case_gdc_id	string	Optional.
disease_code	string	Optional.
endpoint_type	string	Optional.
gender	string	Optional.
hist_subtype	string	Optional.
histology	string	Optional.
program_name	string	Optional.
project_short_name	string	Optional.
sample_barcode	string	Optional.
sample_gdc_id	string	Optional.
sample_type	string	Optional.
site_primary	string	Optional.
source	string	Optional.
summary_file_count	integer	Optional.
summary_file_count_gte	integer	Optional.
summary_file_count_lte	integer	Optional.

Response

If successful, this method returns a response body with the following structure:

```
{
  "case_count": integer,
  "cases": [string],
  "sample_count": integer,
  "samples": [string]
}
```

Parameter name	Value	Description
case_count	integer	Number of cases in the cohort.
cases[]	list	List of cases barcodes in the cohort.
sample_count	integer	Number of samples in the cohort.
samples[]	list	List of sample barcodes in the cohort.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cohorts().create()

Creates and saves a cohort. Takes a JSON object in the request body to use as the cohort's filters. Authentication is required. Returns information about the saved cohort, including the number of cases and the number of samples in that cohort.

Example:

```
python isb_curl.py "https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_
↪api/v3/cohorts/create?name={COHORT NAME}" -H "Content-Type: application/
↪json" -d '{"program_short_name": ["CCLE-BLCA", "CCLE-LUSC"], "gender":
↪"Male"}'
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
from oauth2client.client import OAuth2WebServerFlow
from oauth2client import tools
from oauth2client.file import Storage
import httplib2
import os

CLIENT_ID = '907668440978-0ol0griu70qkeb6k3gnn2vipfa5mgl60.apps.
↪googleusercontent.com'
CLIENT_SECRET = 'To_WJH7-1V-TofhNGcEqmEYi'
EMAIL_SCOPE = 'https://www.googleapis.com/auth/userinfo.email'
DEFAULT_STORAGE_FILE = os.path.join(os.path.expanduser('~'), '.isb_
↪credentials')


def get_credentials():
        oauth_flow_args = ['--noauth_local_webserver']
        storage = Storage(DEFAULT_STORAGE_FILE)
        credentials = storage.get()
        if not credentials or credentials.invalid:
                flow = OAuth2WebServerFlow(CLIENT_ID, CLIENT_SECRET, EMAIL_
↪SCOPE)
                flow.auth_uri = flow.auth_uri.rstrip('/') + '?approval_
↪prompt=force'
                credentials = tools.run_flow(flow, storage, tools.argparser.
↪parse_args(oauth_flow_args))
        return credentials


def get_authorized_service():
        api = 'isb_cgc_ccle_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        credentials = get_credentials()
```

(continues on next page)

```
        http = credentials.authorize(httplib2.Http())
        if credentials.access_token_expired or credentials.invalid:
                credentials.refresh(http)
        authorized_service = build(api, version,␣
→discoveryServiceUrl=discovery_url, http=http)
        return authorized_service


service = get_authorized_service()
body = {'program_short_name': ['CCLE-BLCA', 'CCLE-LUSC'], 'gender': 'Male'}
data = service.cohorts().create(name=name, body=body).execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/
→cohorts/create?name=Cohort_test&fields=case_count%2Cfilters%2Cid%2Clast_
→date_saved%2Cname%2Csample_count
```

{

    "Common": {

        "project_short_name": ["CCLE-BLCA", "CCLE-LUSC"

        ]

    }, "Clinical": {

        "gender": ["Male"

        ]

    }

}

Parameters

Parameter name	Value	Description
name	string	Required.

Request body

In the request body, supply a metadata resource with the following properties:

```
{
  "case_barcode": [string],
  "case_gdc_id": [string],
  "disease_code": [string],
  "endpoint_type": [string],
  "gender": [string],
  "hist_subtype": [string],
  "histology": [string],
  "program_name": [string],
  "project_short_name": [string],
  "sample_barcode": [string],
  "sample_gdc_id": [string],
  "sample_type": [string],
```

```
    "site_primary": [string],
    "source": [string],
    "summary_file_count": [integer],
    "summary_file_count_gte": integer,
    "summary_file_count_lte": integer
}
```

Parameter name	Value	Description
case_barcode[]	list	Optional.
case_gdc_id[]	list	Optional.
disease_code[]	list	Optional. Possible values include: 'BLCA', 'BRCA', 'CESC', 'COAD', 'DLBC', 'ESCA', 'HNSC', 'KIRC', 'LCLL', 'LGG', 'LIHC', 'LUSC', 'MESO', 'MM', 'OV', 'PAAD', 'PRAD', 'SARC', 'SKCM', 'STAD', 'THCA', 'UCEC'.
endpoint_type[]	list	Optional. Possible values include: 'legacy'.
gender[]	list	Optional. Possible values include: 'F', 'M', 'U'.
hist_subtype[]	list	Optional. Possible values include: 'acute_lymphoblastic_B_cell_leukaemia', 'acute_lymphoblastic_T_cell_leukaemia', 'acute_myeloid_leukaemia', 'adenocarcinoma', 'adult_T_cell_lymphoma-leukaemia', 'alveolar', 'anaplastic_carcinoma', 'anaplastic_large_cell_lymphoma', 'astrocytoma', 'astrocytoma_Grade_III', 'astrocytoma_Grade_IV', 'blast_phase_chronic_myeloid_leukaemia', 'Brenner_tumour', 'bronchioloalveolar_adenocarcinoma', 'Burkitt_lymphoma', 'B_cell_lymphoma_unspecified', 'carcinosarcoma-malignant_mesodermal_mixed_tumour', 'chronic_lymphocytic_leukaemia-small_lymphocytic_lymphoma', 'chronic_myeloid_leukaemia', 'clear_cell_carcinoma', 'clear_cell_renal_cell_carcinoma', 'dedifferentiated', 'diffuse_adenocarcinoma', 'diffuse_large_B_cell_lymphoma', 'ductal_carcinoma', 'embryonal', 'endometrioid_carcinoma', 'essential_thrombocythaemia', 'follicular_carcinoma', 'giant_cell_tumour', 'gliosarcoma', 'granulosa_cell_tumour', 'hepatoblastoma', 'hepatocellular_carcinoma', 'Hodgkin_lymphoma', 'intestinal_adenocarcinoma', 'large_cell_carcinoma', 'mantle_cell_lymphoma', 'medullary_carcinoma', 'metaplasia', 'metaplastic_carcinoma', 'mixed_adenosquamous_carcinoma', 'mixed_carcinoma', 'mucinous_carcinoma', 'mucoepidermoid_carcinoma', 'mycosis_fungoides-Sezary_syndrome', 'non_small_cell_carcinoma', 'NS', 'oligodendroglioma', 'papillary_carcinoma', 'papilloma', 'peripheral_T_cell_lymphoma_unspecified', 'plasma_cell_myeloma', 'renal_cell_carcinoma', 'serous_carcinoma', 'signet_ring_adenocarcinoma', 'small_cell_adenocarcinoma',

Response

If successful, this method returns a response body with the following structure:

```
{
  "case_count": integer,
  "filters": [
    {
      "name": string,
      "value": string
    }
  ],
  "id": string,
  "last_date_saved": string,
  "name": string,
  "sample_count": integer
}
```

Parameter name	Value	Description
case_count	integer	Number of unique case barcodes in the cohort.
filters[]	list	List of filters applied to create cohort, if any.
filters[].name	string	Names of filtering parameters used to create the cohort.
filters[].value	string	Values of filtering parameters used to create the cohort.
id	string	Cohort id.
last_date_saved	string	Last date the cohort was saved.
name	string	Name of cohort.
sample_count	integer	Number of unique sample barcodes in the cohort.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cases().get()

Returns information about a specific case, including a list of samples and aliquots derived from this case. Takes a case barcode (*eg* ACC-MESO-1) as a required parameter. User does not need to be authenticated.

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/
↪cases/1034
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
```

```
        api = 'isb_cgc_ccle_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
→api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
→http=httplib2.Http())

service = get_unauthorized_service()
data = service.cases().get(case_barcode='1034').execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/
→cases/{case_barcode}
```

Parameters

Parameter name	Value	Description
case_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure(CCLE does not have clinical data hence the clinical data would remain blank):

```
{
  "aliquots": [string],
  "clinical_data": {},
  "samples": [string]
}
```

Parameter name	Value	Description
aliquots[]	list	List of barcodes of aliquots taken from this patient.
clinical_data	nested object	The clinical data about the patient.
samples[]	list	List of barcodes of samples taken from this patient.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

cases().get_list()

Given a list of case barcodes (*eg* ACC-MESO-1), returns information about them, including a list of samples and aliquots derived from this case. Takes a list of case barcodes (*eg* ACC-MESO-1) as a required data payload. User does not need to be authenticated.

Example:

```
curl --data '{"case_barcodes": ["ACC-MESO-1","Saos-2"]}' https://api-dot-isb-
↪cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/cases
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_ccle_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
↪http=httplib2.Http())

service = get_unauthorized_service()
data = service.cases().get(body={"case_barcodes": ["Saos-2","ACC-MESO-1"]}).
↪execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/
↪cases
```

{

   "case_barcodes": ["ACC-MESO-1", "Saos-2"

   ]

}

Parameters

Parameter name	Value	Description
widths 50	10	50
case_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure(CCLE does not have clinical data hence the clinical data would remain blank):

```
{
  "aliquots": [string],
  "clinical_data": {},
  "samples": [string],
  "sample_barcode": string
}
```

Parameter name	Value	Description
aliquots[]	list	List of barcodes of aliquots taken from this patient.
clinical_data	nested object	The clinical data about the patient.
samples[]	list	List of barcodes of samples taken from this patient.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

samples().cloud_storage_file_paths()

Takes a sample barcode as a required parameter and returns cloud storage paths to files associated with that sample.

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/
↪samples/CCLE-LS1034/cloud_storage_file_paths
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_ccle_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
↪http=httplib2.Http())

service = get_unauthorized_service()
data = service.samples().cloud_storage_file_paths(sample_barcode='CCLE-LS1034
↪').execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/
↪samples/{sample_barcode}/cloud_storage_file_paths
```

Parameters

Parameter name	Value	Description
analysis_workflow_type	string	Optional.
data_category	string	Optional.
data_format	string	Optional.
data_type	string	Optional.
experimental_strategy	string	Optional.
genomic_build	string	Optional.
platform	string	Optional.
sample_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure:

```
{
  "cloud_storage_file_paths": [string],
  "count": integer
}
```

Parameter name	Value	Description
cloud_storage_file_paths[]	list	List of Google Cloud Storage paths of files associated with the cohort.
count	integer	Number of Google Cloud Storage paths returned for the cohort.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

samples().get()

Given a sample barcode (*eg* CCLE-ACC-MESO-1), this endpoint returns all available "biospecimen" information about this sample, the associated case barcode, a list of associated aliquots, and a list of "data_details" blocks describing each of the data files associated with this sample

Example:

```
curl https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/
↪samples/CCLE-LS1034
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```
from googleapiclient.discovery import build
import httplib2


def get_unauthorized_service():
        api = 'isb_cgc_ccle_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
↪api, version)
```

(continues on next page)

```
        return build(api, version, discoveryServiceUrl=discovery_url,
→http=httplib2.Http())

service = get_unauthorized_service()
data = service.samples().get(sample_barcode='CCLE-LS1034').execute()
```

Request

HTTP request:

```
GET https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/
→samples/{sample_barcode}
```

Parameters

Parameter name	Value	Description
analysis_workflow_type	string	Optional.
data_category	string	Optional.
data_format	string	Optional.
data_type	string	Optional.
endpoint_type	string	Optional.
experimental_strategy	string	Optional.
platform	string	Optional.
sample_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure(CCLE does not have any biospecimen data hence it would remain blank):

```
{
  "aliquots": [string],
  "biospecimen_data": {},
  "case_barcode": string,
  "case_gdc_id": string,
  "data_details": [
    {
      "access": string,
      "analysis_workflow_type": string,
      "data_category": string,
      "data_format": string,
      "data_type": string,
      "disease_code": string,
      "endpoint_type": string,
      "experimental_strategy": string,
      "file_gdc_id": string,
      "file_name": string,
      "file_name_key": string,
      "file_size": string,
      "index_file_name": string,
      "platform": string,
      "program_name": string,
      "project_short_name": string,
      "sample_barcode": string,
      "sample_gdc_id": string,
```

```
        "sample_type": string
      }
  ],
  "data_details_count": integer
}
```

Table 11: :header: "**Parameter name**", "**Value**", "**Description**" :widths: 50, 10, 50

aliquots[]	list	List of barcodes of aliquots taken from this participant.
biospecimen_data	nested object	Biospecimen data about the sample.
case_barcode	string	Barcode of the case.
case_gdc_id	string	Case gdc id.
data_details[]	list	List of information about each file associated with the sample barcode.
data_details[].access	string	An indication of the security protocol necessary to fulfill in order to access the data from the file, e.g. open, controlled.
data_details[].analysis_workflow_type	string	The type of workflow used to generate the data file, e.g. 'BWA-aln', 'STAR 2-Pass', 'BWA with Mark Duplicates and Cocleaning'
data_details[].data_category	string	The higher level categorization of the data_type in the file, e.g. 'Biospecimen', 'Clinical', 'Raw sequencing data', 'Simple nucleotide variation'
data_details[].data_format	string	The format of the data file, e.g. 'BAM', 'BCR XML', 'TXT'
data_details[].data_type	string	Data type stored in Google Cloud Storage, e.g. 'Clinical Supplement', 'Biospecimen Supplement', 'Aligned reads', 'Genotypes', 'Diagnostic image'
data_details[].disease_code	string	The disease abbeviation, e.g. 'ACC', 'UVM', 'ALL', 'WT'
data_details[].endpoint_type	string	The GDC files API the data file information was gotten from, e.g. 'legacy', 'current'
data_details[].experimental_strategy	string	The sequencing, array or other strategy used to generate the data file, e.g. 'RNA-Seq', 'WGS', 'Genotyping array'
data_details[].file_gdc_id	string	The GDC assigned id for the file
data_details[].file_name	string	Name of the datafile stored on the GDC system.
data_details[].file_name_key	string	Google Cloud Storage path to file.
data_details[].file_size	string	The size of the file
data_details[].index_file_name	string	For BAM files, the name of its index file
data_details[].platform	string	The sequencing or array platform used, e.g. Illumina HiSeq, Ion Torrent PGM, Affymetrix SNP Array 6.0.
data_details[].program_name	string	The program for which the data was generated, e.g. 'CCLE', 'TARGET','TCGA'.
data_details[].project_short_name	string	The id of the project, e.g. 'CCLE-ACC', 'CCLE-UVM', 'TARGET-ALL-P1', ' TARGET-WT', 'TCGA-ACC', 'TCGA-UVM'
data_details[].sample_barcode	string	Sample barcode.
data_details[].sample_gdc_id	string	The GDC assigned id for the sample
data_details[].sample_type	string	The sample type, e.g. '01', '10', '11'
data_details_count	integer	Number of files associated with the sample barcode.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

samples().get_list()

Given a list of sample barcodes (of length 16, *eg* CCLE-ACC-MESO-1), this endpoint returns the associated case barcode, a list of associated aliquots, and a list of "data_details" blocks describing each of the data files associated with this sample

Example:

```
curl --data '{"sample_barcodes": ["CCLE-LS1034","CCLE-ACC-MESO-1"]}' https://
→api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/samples
```

API explorer example:

Click here to see this endpoint in Google's API explorer.

Python API Client Example:

```python
from googleapiclient.discovery import build
import httplib2

def get_unauthorized_service():
        api = 'isb_cgc_ccle_api'
        version = 'v3'
        site = 'https://api-dot-isb-cgc.appspot.com'
        discovery_url = '%s/_ah/api/discovery/v1/apis/%s/%s/rest' % (site,
→api, version)
        return build(api, version, discoveryServiceUrl=discovery_url,
→http=httplib2.Http())

service = get_unauthorized_service()
data = service.samples().get(body={"case_barcodes": ["CCLE-LS1034","CCLE-ACC-
→MESO-1"]}).execute()
```

Request

HTTP request:

```
POST https://api-dot-isb-cgc.appspot.com/_ah/api/isb_cgc_ccle_api/v3/ccle/
→samples
```

{

   "sample_barcodes": ["CCLE-LS1034", "CCLE-ACC-MESO-1"

   ]

}

Parameters

Parameter name	Value	Description
widths 50	10	50
analysis_workflow_type	string	Optional.
data_category	string	Optional.
data_format	string	Optional.
data_type	string	Optional.
endpoint_type	string	Optional.
experimental_strategy	string	Optional.
platform	string	Optional.
sample_barcode	string	Required.

Response

If successful, this method returns a response body with the following structure(CCLE does not have any biospecimen data hence it would remain blank):

```
{
  "aliquots": [string],
  "biospecimen_data": {},
  "case_barcode": string,
  "sample_barcode": string,
  "case_gdc_id": string,
  "data_details": [
    {
      "access": string,
      "analysis_workflow_type": string,
      "data_category": string,
      "data_format": string,
      "data_type": string,
      "disease_code": string,
      "endpoint_type": string,
      "experimental_strategy": string,
      "file_gdc_id": string,
      "file_name": string,
      "file_name_key": string,
      "file_size": string,
      "index_file_name": string,
      "platform": string,
      "program_name": string,
      "project_short_name": string,
      "sample_barcode": string,
      "sample_gdc_id": string,
      "sample_type": string
    }
  ],
  "data_details_count": integer
}
```

Table 12: :header: "**Parameter name**", "**Value**", "**Description**" :widths: 50, 10, 50

aliquots[]	list	List of barcodes of aliquots taken from this participant.
biospecimen_data	nested object	Biospecimen data about the sample.
case_barcode	string	Barcode of the case.
case_gdc_id	string	Case gdc id.
data_details[]	list	List of information about each file associated with the sample barcode.
data_details[].access	string	An indication of the security protocol necessary to fulfill in order to access the data from the file, e.g. open, controlled.
data_details[].analysis_workflow_type	string	The type of workflow used to generate the data file, e.g. 'BWA-aln', 'STAR 2-Pass', 'BWA with Mark Duplicates and Cocleaning'
data_details[].data_category	string	The higher level categorization of the data_type in the file, e.g. 'Biospecimen', 'Clinical', 'Raw sequencing data', 'Simple nucleotide variation'
data_details[].data_format	string	The format of the data file, e.g. 'BAM', 'BCR XML', 'TXT'
data_details[].data_type	string	Data type stored in Google Cloud Storage, e.g. 'Clinical Supplement', 'Biospecimen Supplement', 'Aligned reads', 'Genotypes', 'Diagnostic image'
data_details[].disease_code	string	The disease abbeviation, e.g. 'ACC', 'UVM', 'ALL', 'WT'
data_details[].endpoint_type	string	The GDC files API the data file information was gottern from, e.g. 'legacy', 'current'
data_details[].experimental_strategy	string	The sequencing, array or other strategy used to generate the data file, e.g. 'RNA-Seq', 'WGS', 'Genotyping array'
data_details[].file_gdc_id	string	The GDC assigned id for the file
data_details[].file_name	string	Name of the datafile stored on the GDC system.
data_details[].file_name_key	string	Google Cloud Storage path to file.
data_details[].file_size	string	The size of the file
data_details[].index_file_name	string	For BAM files, the name of its index file
data_details[].platform	string	The sequencing or array platform used, e.g. Illumina HiSeq, Ion Torrent PGM, Affymetrix SNP Array 6.0.
data_details[].program_name	string	The program for which the data was generated, e.g. 'CCLE', 'TARGET','TCGA'.
data_details[].project_short_name	string	The id of the project, e.g. 'CCLE-ACC', 'CCLE-UVM', 'TARGET-ALL-P1', ' TARGET-WT', 'TCGA-ACC', 'TCGA-UVM'
data_details[].sample_barcode	string	Sample barcode.
data_details[].sample_gdc_id	string	The GDC assigned id for the sample
data_details[].sample_type	string	The sample type, e.g. '01', '10', '11'
data_details_count	integer	Number of files associated with the sample barcode.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.6.4 Using Google Compute Engine

For those ISB-CGC users whose research goals require the ability to run large compute jobs, all of the power and infrastructure behind Google Compute (Compute Engine, Container Engine, Dataproc, and Dataflow) and Google Genomics, are at your disposal.

Our goal is to help you assemble the tools and data (TCGA data, your data, reference data, *etc*) that you need to answer your research questions in the most efficient and cost-effective way possible.

Towards that end, we have created a github repository called examples-Compute with examples to get you started. This repository will continue to grow and we welcome your contributions and suggestions. You can also find a number of useful recipes in the Google Genomics Cookbook, also here on readthedocs.

For an introduction to using Google Compute Engine, please follow the link below.

Introduction to Google Compute Engine

Google Compute Engine (GCE) is the Infrastructure as a Service (IaaS) component of Google Cloud Platform (GCP). GCE offers scale, performance and value, letting you easily create and run virtual machines (VMs) on Google infrastructure.

We have tried to put together some basic documentation for ISB-CGC users who are new to the Google Cloud Platform, but your main source of information should generally be the official Google Cloud Platform documentation. We have found that sometimes the wealth of available information can result in information overload, so we hope that this brief introduction will be useful to you. If you are still feeling lost, please let us know and we'll do our best to get you pointed in the right direction.

Setting up your GCP project

This setup guide assumes that you are already a member of a GCP project with either "Owner" or "Editor" rights. If you need a GCP project, you may request one as part of the ISB-CGC community evaluation phase going on now.

Google Cloud Console

If you are new to the Google Cloud, it is a good idea to become familiar with the Cloud Console (which we will generally refer to simply as the Console). You can get help from within the Console by clicking on the Help (question mark) icon near the upper right-hand corner. The Console provides a convenient web UI for managing resources within your cloud project, and can be useful for obtaining a quick, high-level snapshot of the state of your project. The "Home" page will list, for example, the number of buckets you have created in Cloud Storage, the number of datasets in BigQuery, and the number of VMs you have running under App Engine or Compute Engine. It also shows the charges incurred by this project so far this month.

Enable the Compute Engine API

The Compute Engine API is probably enabled by default on your GCP project, but you can verify this through the Console: click on the menu icon in the upper left hand corner (when you hover over it you will see "Products and services"), and then select the API Manager. The API Manager page has two sections: Overview and Credentials. Within the Overview page, you can see a list of all "Google APIs" and a list of the "Enabled APIs".

You can check your list of "Enabled APIs", or simply select the "Compute Engine API" link which should be at the very top of the list of "Popular APIs". Once you are on the "Google Compute Engine" page, you should either see a blue button with the word "Enable" or a white "Disable button. If the button says Enable, click on it. This process will take a minute or two, after which you will be prompted to "Go to Credentials". You should not need to create new credentials at this time – you will typically be using Application Default Credentials. (This blog post introducing Application Default Credentials may also be helpful.) The proper use of credentials is frequently one of the most complicated aspects of interacting with the Google Cloud Platform. If you are having problems, please let us know.

You may also find the official Compute Engine Getting Started Guide helpful.

Google Cloud SDK

Depending on how you choose to interact with the Google Cloud Platform, you may want to install the Google Cloud SDK on your local workstation. The Google Cloud SDK is a set of command-line interface (CLI) tools that you can use to manage resources and applications hosted on GCP. (Note that components of the the SDK are updated quite frequently. You will be notified when updates are available anytime you use one of the SDK tools. The command will still run, but you will be notified that "Updates are available for some Cloud SDK components" and you will be given instructions on how to update your local copy of the SDK.)

Confirm that you have installed the SDK and have access to it by typing `gcloud --version` at the command line of your own linux workstation or from the Cloud Shell (for more details about the Cloud Shell, see the next section). You should see something like this:

```
Google Cloud SDK 98.0.0

bq 2.0.18
bq-nix 2.0.18
core 2016.02.22
core-nix 2016.02.05
gcloud
gsutil 4.16
gsutil-nix 4.15
```

Google Cloud Shell

Google Cloud Shell provides you with command-line access to computing resources hosted on GCP is available from the Console. Cloud Shell provides you with a temporary VM running a Debian-based Linux OS, with 5 GB of persistent disk storage per user, and the Google Cloud SDK and other tools pre-installed.

From the Console, you will find the icon for the Cloud Shell in the top-most blue bar, near the right-hand corner, between your GCP project name and the "Send feedback" icon. If you click on that icon (the hover-card should read "Activate Google Cloud Shell"), it will take a minute or two for you VM to be provisioned, after which you will see a prompt saying "Welcome to Cloud Shell" in the new window that has appeared at the bottom of your Console page. You can "pop" that window out of your browser page by clicking on the "Open in new window" icon in the upper right-hand corner of the shell window.

Authenticate with Google

Regardless of how you choose to interact with the Google Cloud, you will need to authenticate yourself. How this authentication takes place will depend on "where" you are. If you have signed into Chrome using your Google identity and you then go to the Console, you will already have been authenticated. If you are at the Linux prompt of the Cloud Shell, you have also already been authenticated because that Shell (and that VM) were launched for you from your Console. If you are at the Linux prompt of your local workstation, you will need to authenticate using the **gcloud** command line utility.

There are two approaches:

- gcloud init launches an interactive Getting Started workflow for gcloud;
- gcloud auth login obtains access credentials for your user account via a web-based authorization flow.

These approaches may ask you to cut-and-paste a long URL into a browser, sign in using your Google credentials, click "Allow" to allow Google to access certain information about you, and may also ask that you cut-and-paste an authorization token from your browser back into the Linux shell.

Once you have authenticated, you can see information about your current configuration by typing `gcloud config list`. You can set additional properties using the `gcloud config set` command. The most common properties you are likely to want to verify (list), or set explicitly are:

- account
- project
- compute/region
- compute/zone
- container/cluster

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Launching a Virtual Machine (VM)

You can launch a virtual machine (which we will generally refer to as a VM) from the Console or from the command line using the Google Cloud SDK. We will describe both of these approaches here.

You should already be somewhat familiar with the Console, and hopefully you have tried invoking the **gcloud** command from your command-line. The **gcloud** command-line tool can be used to manage both your development workflow and your GCP resources. (For more details, please look at the official gcloud Tool Guide.)

Bundled into the gcloud CLI are several **commands** and **groups** of sub-commands. The group of sub-commands that allows you to read and manipulate GCE resources is gcloud compute

Launch a VM using the Console

After you have enabled the Compute Engine API for you project, you can go the Compute Engine section of the Console. (Select the menu icon in the far upper-left corner, and then choose "Compute Engine" from the flyout panel.) The first time, you may need to wait a minute or so while "Compute Engine is getting ready."

You will now be on the "VM instances" page. (There are may other pages that are accessible from the left side-panel.) The first time you visit this page, you will see two options: "Create Instance" or "Take the quickstart". After the first time, you may see a different page with a list of existing (running or stopped) VMs with a CPU utilization graph. At the top of this page, you will see options to "CREATE INSTANCE", "CREATE INSTANCE GROUP", "RESET", "START", "STOP", and "DELETE" VM instances.

After selecting the "Create Instance" option, you will be sent to the "Create an instance" page where defaults will be selected for the Name, Zone, Machine type, etc:

- Name: this name is relatively arbitrary, choose something that is meaningful to you;
- Zone: choose one of the us-east or us-central zones;
- Machine type: you can specify a VM with anywhere between 1 and 16 cores (aka vCPUs), and with up to 100 GB of RAM (you can try the "Customize" view if you prefer a more graphical approach); note that as you change the specifications of the VM, the estimated cost shown on this page will update;
- Boot disk: the default boot disk and OS will be shown, but you can change this as you wish: the "Change" button will result in a flyout panel where you can choose from a variety of Preconfigured images (Debian, CentOS, Ubuntu, RedHat, etc) or previously created images or disks; you can also choose between "standard disks" and faster (and more expensive) solid-state drives (SSDs), and specify the size of the disk (up to 64TB).

Other options below the "Management, disk, . . . " line include Preemptibility (default is OFF), Automatic restart (default is ON), and what to do during infrastructure maintenance (default is to "migrate VM" so that you will not experience any downtime).

Once you have all of the options set, you can click on the blue **Create** button. You can also see you could use the REST or command-line interfaces to do perform the exact same option. (The Console is just a friendlier interface between you and more direct REST-based access to the same functionality.)

Creating the VM should take less than a minute, after which you will see it listed on the "VM instances" page, with the Name, Zone, Disk, Network, and External IP address shown. There is also an SSH button that you can use directly from the Console.

Launch a VM using the CLI

The command to create a new GCE VM instance is `gcloud compute instances create`. The complete documentaiton can be found online or by typing `gcloud compute instances create --help` on the command line.

Some defaults can be obtained (if available) from your configuration settings. For example, if you don't want to have to specify the zone of the instances, you can set the compute/zone property, for example: ` gcloud config set compute/zone us-central1-a ` A list of zones can be fetched by running: ` gcloud compute zones list `

Here is a very simple command to create a VM: ` gcloud compute instances create my-instance --machine-type g1-small `

Accessing your new VM

Whether you have created your VM from the Console or using the gcloud CLI, you can find it and ssh to it, again using either the Console or the CLI:

- From the Console, go to Compute Engine > VM instances, and then click on the **SSH** button on the far-right of the row describing the specific VM you would like to connect to.
- Using the CLI, simply use the command `gcloud cmopute ssh` followed by the instance name.

Shutting down your VM

Remember that as long as your VM is running, whether or not *you* are actually doing anything with it, charges will be incurred. It is therefore a good idea to get in the habit of shutting down VMs as soon as you are finished with your work. They can easily be restarted an hour, day, or week later. Note that resources that are *attached* to a stopped VM (such as persistent disks) *will*, however continue to incur charges. Compared to the cost of the VM, though, the cost of a persistent disk is typically negligible: a 50 GB standard persistent disk only costs $2 per month, and 1 TB costs $40.

If you know that you won't never need this specific VM again, or you don't want to continue paying for the persistent disk, or you would rather start a fresh VM with an updated OS next time, then you can go ahead and **delete** the VM rather than just stopping it.

From the command-line, the relevant commands are `gcloud compute instances stop` and `gcloud compute instances delete`.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Creating and Managing Persistent Disks

As described in the previous section, you can specify the boot disk when launching a VM from the Console and from the command-line. There are times when you may want to create and attach additional disks to an instance. There are three main steps in this process: you must first **create** the disk, then you must **attach** it to the instance, and finally you must **format** it. When you are finished, you may want to **detach** the disk and when you are done with it, you will want to **delete** it. We will describe each of these steps in a bit more detail below. You may also want to see the Google documentation on Adding Persistent Disks.

Create a Persistent Disk

The **gcloud** command for creating a persistent disk is `gcloud compute disks create`. The most common options you'll probably use are `--size`, `--type`, and `--zone` (see this page for more details). For example:

```
gcloud compute disks create disk-1 --size 500GB
```

will create a 500 GB disk named "disk-1", using default settings (*eg* the type will be pd-standard).

Attach a Persistent Disk

The **gcloud** command to attach a newly created disk to a previously created instance looks like this:

> gcloud compute instances attach-disk –disk disk-1 –device-name my-instance

Note that this command is part of the **gcloud compute instances** group rather than the **gcloud compute disks** group. Details about additional options can be found in the documentation. For example the default mode is **rw** (read-write), but you can also specify that a disk be attached **ro** (read-only).

Format a Persistent Disk

In order to format a disk that you've attached to an instance, you need to first log on to that instance:

```
gcloud compute ssh  my-instance
```

For complete details, please refer to the Google documentation on formatting and mounting non-root persistent disks; but there are two main steps: first you must format the disk using the **mkfs** tool (note that this will delete any *existing* data on the disk), and second you must use the **mount** tool to mount the disk at a specified mount-point:

```
sudo mkfs.ext4 -F /dev/disk/by-id/disk-1
sudo mkdir /mnt/pd1
sudo mount -o discard,defaults /dev/disk/by-id/disk-1 /mnt/pd1
```

Detach a Persistent Disk

Detaching a disk is a two step process: first you unmount the disk (using the **umount** command, *from* the instance to which it is attached), and then (after logging out from that instance) you use the **gcloud** tool:

```
$sudo umount /dev/disk/by-id/disk-1
$exit

gcloud compute instances detach-disk my-instance --disk disk-1
```

Delete a Persistent Disk

Note that a boot disk will be deleted if you delete the instance that it is attached to (as long as the auto-delete property for the disk was set to "yes" (the default) when it was created). In all other cases, you will need to delete the disk manually using the `gcloud compute disks delete` command. Note that disks can be deleted only if they are *not* being used by any VM instances.

You can also see and manage persistent disks from the Console on the Compute Engine > Disks page.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.6.5 Running Workflows

In the last few years there has been significant renewed interest in using a workflow approach to perform data analysis on Genomic, RNASeq, Proteomic and other types of data driven by the growth in big data and the popularity of cloud technologies ability for scalable storage and computing. The concept of a workflow was defined about 20 years ago by the Workflow Management Coalition as: *"The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules."* The focus of this particular organization is on something called BPM or *Business Process Management*.

Scientific workflows have emerged to manage and describe the complexity that arises in scientific experiments, as well as data analysis and data processing. Complex workflows are created by linking or *chaining* several components or tasks into a *pipeline*.

A complete scientific workflow *system* requires first a clearly defined *language* and *grammar* which can be used to describe a workflow. Given a clearly specified workflow, a "workflow runner" of some sort is necessary in order to be able to actually *run* the workflow. A "runner" generally implements the following "roles": a master or administrator, a scheduler, a task executor, and workers: in which the master receives and parses workflow document(s) and communicates requirements to the scheduler; the scheduler is typically trying to optimize usage of the available workers based on the requirements of the master(s), the executor causes tasks to be run on the specified schedule, and the workers do the work.

Out of this demand for scientific workflows have emerged sseveral competing description languages such as the CWL (Common Workflow Language), WDL (Workflow Description Language), and NextFlow to name a few. These languages underwent rapid development and changes in the beginning and now are becoming more stable for use in production environments. Coupled with these languages are runner implementations such as cwltool, rabix, toil, and cromwell and new API standards such as GA4GH's Task Execution Schema (TES) and the Workflow Execution Schema (WES). With the advent of so many choices; all under rapid development and at various stages of completeness its challenging to make a choice on any one technology. ISB CGC's approach therefore has been not to make a choice but to instead enable as many of these technologies as possible through documentation, support, and where necessary infrastructure.

For more information on workflows see:

The Common Workflow Language (CWL)

The Common Workflow Language (CWL) is emerging as a standard for defining and sharing bioinformatic workflows, and the NCI-GDC is planning to release all of its standardized workflows in this format.

In the sections below, we present a tutorial on running a sample NCI-GDC workflow with step-by-step instructions to run it on a sample input BAM file using a Google Compute Engine (GCE) VM. The second section describes how to use a convenient "helper-script" called cwl_runner (available on github) which wraps many of the individual steps required to create a GCE disk, spin up a GCE VM, mount and format the disk, *etc*, allowing you to run a CWL workflow in one easy step.

Running the NCI-GDC DNA-Seq workflow

In this section, we will guide you through the steps to run the NCI-GDC's DNA-Seq harmonization workflow on Google Compute Engine. This workflow is available on github here. The instructions here are based on the NCI-GDC's README and have been customized to run on GCE.

1. Create a GCE VM with Disk

From the Cloud Console > Compute Engine > VM instances page click on **[+] CREATE INSTANCE**, and:

- set Name (*eg* cwl-test-1)
- set Zone (*eg* us-central1-c)
- set Machine type (*eg* 4 vCPUs with 15 GB memory)
- Change the boot disk to Ubuntu 14.04 LTS with 10 GB standard persistent disk (note that the boot disk will be named the same as the VM, *ie* cwl-test-1)
- leave the Identity and API access box as is (with "Compute Engine default service account" and "Allow default access" selected)
- expand the "Management, disk, networking, SSH keys section":
    - select the Disks tab
    - click on **+ Add item**
    - in the Name pull-down, select "Create disk": a "Create a disk" panel will open:
        * set Name (*eg* cwl-disk-1) – do not use the same name as the VM!
        * set Source type to "None (blank disk)"
        * set Size (*eg* 500 GB)
        * leave default Encryption (which is "Automatic (recommended)")
        * click on the blue **Create** button – this will create the disk only at this time
    - before clicking on the **Create** button (for the VM), click on the bottom line where it says "Equivalent REST or command line" – you can save this command-line and re-use it later to create the same VM from the command-line rather than repeating this interactive process; it is also a nice record of exactly how the VM was created
    - now click on the **Create** button – you will see the VM "spinning up" on the VM instances page

Example command-line equivalents to create the disk and the VM (you will need to substitute in your own Google Cloud Platform (GCP) project:

```
$ gcloud compute --project <YOUR-PROJECT-ID> disks create "cwl-disk-1" --size "500" --
↪zone "us-central1-c" --type "pd-standard"

$ gcloud compute --project <YOUR-PROJECT-ID> instances create "cwl-test-1" --zone "us-
↪central1-c" --machine-type "n1-standard-4" --network "default" --maintenance-policy
↪"MIGRATE" --scopes default="https://www.googleapis.com/auth/devstorage.read_only",
↪"https://www.googleapis.com/auth/logging.write","https://www.googleapis.com/auth/
↪monitoring.write","https://www.googleapis.com/auth/servicecontrol","https://www.
↪googleapis.com/auth/service.management.readonly","https://www.googleapis.com/auth/
↪trace.append" --disk "name=cwl-disk-1,device-name=cwl-disk-1,mode=rw,boot=no" --
↪image "/ubuntu-os-cloud/ubuntu-1404-trusty-v20161205" --boot-disk-size "10" --boot-
↪disk-type "pd-standard" --boot-disk-device-name "cwl-test-1"
```

2. Configure the VM

Now you can ssh to your VM from any command-line where you have the cloud SDK installed. If you don't have the cloud SDK installed on your local machine, you can use the Cloud Shell directly from your browser in the Cloud Console.

```
$ gcloud compute --project <YOUR-PROJECT-ID> ssh --zone "us-central1-c" "cwl-test-1"
```

2.1 Install Packages

Use the following commands to install the necessary packages:

```
$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-keys
↪58118E89F3A912897C070ADBF76221572C52609D

$ echo "deb https://apt.dockerproject.org/repo ubuntu-trusty main" | sudo tee /etc/
↪apt/sources.list.d/docker.list

$ sudo aptitude update

$ sudo aptitude install apt-transport-https ca-certificates docker-engine htop libffi-
↪dev libssl-dev nodejs python-dev virtualenvwrapper
```

After this last command, you will need to respond "Yes" to install the new packages.

2.2 Format and Mount the Disk

You can see the disks that are attached to your VM by using the following command:

```
$ ls /dev/disk/by-id
```

which should respond with something like:

```
google-cwl-disk-1  google-cwl-test-1-part1                    scsi-0Google_
↪PersistentDisk_cwl-test-1
google-cwl-test-1  scsi-0Google_PersistentDisk_cwl-disk-1  scsi-0Google_
↪PersistentDisk_cwl-test-1-part1
```

The first disk listed above (google-cwl-disk-1) is the additional disk that was crated, while the second one (google-cwl-test-1) is the boot disk, with the same name as the VM. The following commands differ slightly from those specified in the NCI-GDC README but the result will be the same:

```
$ sudo mkfs.ext4 -F -E lazy_itable_init=0,lazy_journal_init=0,discard /dev/disk/by-id/
→google-cwl-disk-1
$ sudo mkdir -p /mnt/SCRATCH
$ sudo mount -o discard,defaults /dev/disk/by-id/google-cwl-disk-1 /mnt/SCRATCH
$ sudo chmod 777 /mnt/SCRATCH
```

You can now verify that the disk has been properly mounted using the `df -h` command:

```
$ df -h

File system      Size    Used    Avail    Use%    Mounted on
/dev/sdb         493G     70M     467G      1%    /mnt/SCRATCH
```

and as you can see, close to 500G of space is available mounted as /mnt/SCRATCH.

2.3 Prepare Docker and CWL

These next sets of commands will get you ready to run docker on this VM. You will need to log out and log back in a couple of times to force certain changes to take effect.

```
$ mkdir /mnt/SCRATCH/docker
$ sudo bash -c 'echo DOCKER_OPTS=\"-g /mnt/SCRATCH/docker/\" >> /etc/default/docker'
$ sudo gpasswd -a ${USER} docker
$ sudo service docker restart
$ exit
```

The last command will log you out of your VM, so you will need to log back in using the same `gcloud ssh` command you used before. Once you're back on the VM:

```
$ echo "source /usr/share/virtualenvwrapper/virtualenvwrapper.sh" >> ~/.bashrc
$ exit
```

Sign back in again, and then create a "virtualenv" called "cwl". This will change your command-line prompt to indicate that you are in a new environment:

```
$ mkvirtualenv --python /usr/bin/python2 cwl
(cwl) $
```

A few more install commands and you'll be ready to go:

```
(cwl)$ pip install --upgrade pip
(cwl)$ pip install 'requests[security]' --no-cache-dir
(cwl)$ wget https://github.com/NCI-GDC/cwltool/archive/1.0_gdc_g.tar.gz
(cwl)$ pip install 1.0_gdc_g.tar.gz --no-cache-dir
```

3. Run the DNA-Seq workflow

3.1 Clone the NCI-GDC github repo

You should now be in your home directory, in the (cwl) virtualenv. Clone the NCI-GDC dna-seq-cwl repo:

```
(cwl)$ git clone https://github.com/NCI-GDC/gdc-dnaseq-cwl.git
```

Now you will have a subdirectory called `gdc-dnaseq-cwl` in your home directory, containing the NCI-GDC DNA-Seq harmonization workflow. The main workflow is in the CWL file `~/gdc-dnaseq-cwl/workflows/dnaseq/transform.cwl`.

3.2 Load Reference and Input Data Files

The DNA-Seq workflow requires some reference data files that can be obtained from the NCI-GDC. These include the dbsnp vcf (3 GB), the reference genome (835 MB), and the bwa indexed genome (3.2 GB). (Uploading these to your VM disk should take 5-10 minutes.)

```
(cwl)$ mkdir /mnt/SCRATCH/hg38_reference
(cwl)$ cd /mnt/SCRATCH/hg38_reference
(cwl)$ wget https://gdc-api.nci.nih.gov/data/4ba1c087-ec80-47c4-a9d5-e9bb9933fef4 -O␣
↪dbsnp_144.hg38.vcf.gz
(cwl)$ wget https://gdc-api.nci.nih.gov/data/62f23fad-0f24-43fb-8844-990d531947cf
(cwl)$ tar xvf 62f23fad-0f24-43fb-8844-990d531947cf
(cwl)$ wget https://gdc-api.nci.nih.gov/data/964cbdac-1043-4fae-b068-c3a65d992f6b
(cwl)$ tar xvf 964cbdac-1043-4fae-b068-c3a65d992f6b
```

Finally, let's copy a small example BAM file (300 MB) from the 1000G repository:

```
(cwl)$ cd /mnt/SCRATCH
(cwl)$ wget ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/phase3/data/NA12878/
↪alignment/NA12878.chrom20.ILLUMINA.bwa.CEU.low_coverage.20121211.bam
```

At this point you could also obtain a bam file either from the NCI-GDC or from one of the ISB-CGC Cloud Storage buckets.

3.3 Run DNA-Seq CWL workflow

Now we're ready to run the workflow using the CWL-runner **cwltool**. The input file that we just copied to our VM disk is in `/mnt/SCRATCH/alignment/NA12878.chrom20.ILLUMINA.bwa.CEU.low_coverage.20121211.bam`. Let's create a sub-directory for the processed results:

```
(cwl)$ mkdir /mnt/SCRATCH/NA12878.chrom20.ILLUMINA.bwa.CEU.low_coverage.20121211
(cwl)$ cd /mnt/SCRATCH/NA12878.chrom20.ILLUMINA.bwa.CEU.low_coverage.20121211
(cwl)$ mkdir tmp cache
(cwl)$ nohup cwltool --debug --tmpdir-prefix tmp/ --cachedir cache/ \
        ~/gdc-dnaseq-cwl/workflows/dnaseq/transform.cwl \
        ~/gdc-dnaseq-cwl/workflows/dnaseq/NA12878.chrom20.ILLUMINA.bwa.CEU.low_
↪coverage.20121211.json &
```

While that is running, you can go back to the Cloud Console, to the Compute Engine > VM instances page, and click on the name of this VM. This will take you to a page describing this specific VM, and you can see a trace of CPU utilization, and other metrics.

Let's also take a closer look at the `cwltool` command used above. You can find more details at the cwltool github repo and at commonwl.org. The basic form of the cwltool command is:

```
$ cwltool [tool-or-workflow-description] [input-job-settings]
```

Looking at the way cwltool was invoked above, we see that the `tool-or-workflow-description` is in `~/gdc-dnaseq-cwl/workflows/dnaseq/transform.cwl` and the `input-job-settings` are in `~/gdc-dnaseq-cwl/workflows/dnaseq/NA12878.chrom20.ILLUMINA.bwa.CEU.low_coverage.20121211.json`. Let's have a closer look at those, starting with the smaller input-job-settings JSON document. It defines three objects, each of which is of class "File", with a specified "path", *eg*:

```
"bam_path": {
    "class": "File",
    "path": "/mnt/SCRATCH/NA12878.chrom20.ILLUMINA.bwa.CEU.low_coverage.20121211.bam"
}
```

and it also specifies a "thread_count" value (8), and a "uuid". You can see these inputs defined near the top of the CWL document (transform.cwl).

3.4 Run-time and Compute-costs

This sample task takes about 2 hours to run. The costs associated with running this task are: 2 hours of GCE VM time plus 2 hours of persistent disk time (GCE pricing details), which comes to approximately $0.400 for the VM and $0.056 for the persistent disks, for a total of **$0.456**. (The n1-standard-4 VM chosen above costs $0.200 per hour, and the disk costs, at $0.040 per GB per month for standard provisioned space, were computed as 510 GB x $0.040 per GB per month x 2 hours / 730 hours per month.)

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

The cwl_runner "helper" script

The cwl_runner script can be found on github in the **googlegenomics/pipelines-api-examples** repository, in the cwl_runner folder. You may want to refer to the README file in the github repo, we will also provide an overview and summary of what this script does below, with some additional details that you may find useful.

The basic prerequisites to be able to run this example are:

- you have the necessary privileges to launch a VM in a Google Cloud Project (GCP) project
- you must have an existing Google Cloud Storage (GCS) bucket
- you have the Cloud SDK installed

There are three scripts in the github repo:

- `cwl_runner.sh` is the main bash script which takes care of most of the steps described in this tutorial;
- `cwl_startup.sh` is the VM startup script which will automatically be run as soon as the VM spins up;
- `cwl_shutdown.sh` is the VM shutdown script which takes care of final steps such as copying stdout and stderr to GCS;

The main script, cwl_runner.sh, is the only one that you will invoke directly. It has several different options (which you can learn more about by using the `--help` option), but the only required ones are:

- `--workflow-file PATH`: the absolute path to the CWL workflow document;
- `--settings-file PATH`: the absolute path to the JSON settings file;
- `--output GCS_PATH`: the output location in GCS where all outputs and logs will be copied after the workflow completes.

The script then invokes two `gcloud compute` commands (`gcloud` is part of the cloud SDK:

- `gcloud compute disks create`: to create a persistent disk in the (optionally user-specified) zone, of (optionally user-specified) size;

- `gcloud compute instances create`: to create a virtual machine (VM), in the same zone as the disk, with the previously created disk attached, with the (optionally user-specified) machine type.

If the user specifies, the VM can be a preemptible VM, which can be a good way to minimize compute costs, under the right circumstances.

The other information that the VM needs is passed in as metadata. Metadata is stored as key:value pairs. There is a default set of metadata entries that every VM has access to, and custom metadata can also be set when a VM is created. This metadata will be available to the VM from the metadata server.

The following metadata keys are specified by the cwl_runner script and will be available to the VM:

- startup-script-url

- shutdown-script-url

- operation-id

- workflow-file

- settings-file

- input

- input-recursive

- output

- runner

- status-file

- keep-alive

The cwl_runner script invoked by the user will create a random `OPERATION-ID` and write three script files to the specified output location in GCS:

- `cwl_runner-<OPERATION-ID>.sh`

- `cwl_startup-<OPERATION-ID>.sh`

- `cwl_shutdown-<OPERATION-ID>.sh`

and will run the `gcloud compute disks create` command, followed by the `gcloud compute instances create` command.

In this case, the "startup" script will take care of pretty much everything we want this particular VM to do:

- local bash variables are set based by retrieving the instance metadata from the metadata server

- the disk is mounted and formatted

- folders are created for workflow inputs and outputs

- input files are copied from GCS to local disk

- the workflow and settings files are copied from GCS to local disk

- the executor (cwltool) is invoked to run the workflow

- outputs are copied from local disk out to GCS

- the VM is shut down (using the command `gcloud compute instances delete`) unless the `--keep-alive` option was set

Following the example provided on github, you can invoke cwl_runner from the command-line (anywhere where you have the Cloud SDK installed, with your GCS bucket and optional folder name instead of `MY-BUCKET/my-work-folder` below):

```
./cwl_runner.sh \
  --workflow-file gs://genomics-public-data/cwl-examples/gdc-dnaseq-cwl/workflows/
↪dnaseq/transform.cwl \
  --settings-file gs://genomics-public-data/cwl-examples/gdc-dnaseq-cwl/input/gdc-
↪dnaseq-input.json \
  --input-recursive gs://genomics-public-data/cwl-examples/gdc-dnaseq-cwl \
  --output gs://MY-BUCKET/my-work-folder \
  --machine-type n1-standard-4
```

In this example, the JSON settings file specifies 5 items:

- bam_path (a small ~300MB low-coverage BAM for chromosome 20 only from the 1000G project)
- reference_fasta_path (the GRCh38 reference FASTA file from the GDC Reference Files)
- db_snp_path
- thread_count
- uuid

For more details on machine-types, please see the Google documentation on predefined machine types and if you find that none of those quite fit your requirements you may be interested in using one of the available custom machine types.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

The Workflow Description Language (WDL)

The Workflow Description Language (WDL) which is in use at the Broad Institute, is an alternative to CWL. It is supported by a powerful workflow execution engine called Cromwell, which includes multiple "backends" such as GridEngine, HTCondor, Spark, the Google "Pipelines API" (formerly known as JES, *ie* "Job Execution Service", aka GGP), and the new GA4GH TES (*ie* "Task Execution Service").

We will focus on the two backends that directly support running workflows on Google Compute Engine VMs:

- Google Genomics Pipelines API (formerly known as JES)
- GA4GH Task Execution Service (aka TES)

Google Genomics Pipelines

The so-called "Pipelines API" is a task runner that lets you run a command-line executable in Docker on a Google Compute Engine VM. Since it is truly a "task" runner rather than a full "pipeline" runner, we generally refer to it as GGP so that the usage of the word "pipeline" is not confusing. We also find the additional term "API" unnecessary.

GGP can be "called" using command-line interface (part of the Cloud SDK `gcloud` tool), or as a web service API that can be called programmatically. When using GGP from the command-line, each task is defined in a YAML (or JSON) file.

The Google documentation for the "Genomics Pipelines" can be found here and on readthedocs, and there are numerous easy-to-follow examples on github here.

You can use wdl_runner to run a WDL workflow using Cromwell+GGP on the Google Cloud. Documentation can be found on github and you can run a GATK workflow by following this Google Genomics documentation.

GA4GH Task Execution Service

The GA4GH TES was inspired by GGP, with the broader goal of defining a platform agnostic interface between workflow management systems, schedulers, and workflow language interpreters on the *frontend* of the TES interface, and the actual workes, nodes, VMs, and filesystems on the *backend*. Although this effort started only a few months ago, progress has been rapid and a reference implementation is available on github.

As described in this post over on the WDL Blog, TES has been recently added as a new backend to Cromwell.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

As an alternative to workflows, we also recommend users consider dsub, an command-line tool that uses Docker to make it easy to submit and run single commands or batch scripts in the cloud. The dsub program is loosely based off of more traditional HPC systems such as Grid Engine or PBS where you "submit" your commands to a job scheduler that manages the compute resources, file I/O, and process execution.

More information on dsub and examples of its usage can be found at:

- Google dsub Documentation
- kallisto example
- EricScript example

Lastly, the (now deprecated) ISB-CGC-pipelines framework was developed to facilitate running single step tasks at scale, for example: running FastQC over tens of thousands of FastQ files.

ISB-CGC-pipelines Framework

This framework was built around the Google Genomics Pipelines API (described in more detail below) and is intended to allow you to run single tasks at scale, allowing you to tailor how and when the tasks are submitted, monitor them as they finish etc.

Note: With the advent of Google's dsub the ISB-CGC-pipelines framework is deprecated and no longer supported.

Google Genomics Pipelines

The so-called "Pipelines API" is a task runner that lets you run a command-line executable in Docker on a Google Compute Engine VM. Since it is truly a "task" runner rather than a full "pipeline" runner, we generally refer to it as GGP so that the usage of the word "pipeline" is not confusing. We also find the additional term "API" unnecessary.

GGP can be "called" using command-line interface (part of the Cloud SDK `gcloud` tool), or as a web service API that can be called programmatically. When using GGP from the command-line, each task is defined in a YAML (or JSON) file.

The Google documentation for the "Genomics Pipelines" can be found here and on readthedocs, and there are numerous easy-to-follow examples on github here.

ISB-CGC-pipelines

The ISB-CGC-pipelines source code and documentation is available on github. Detailed documenation is available directly in the README on github, and tutorial slides are also available.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.7 Quick Links

The table below describes the types of functions that can be performed in ISB-CGC and provides links to documentation on how to do those functions.

Function	In GUI	With Command Line
Work on the Google Cloud Platform	• Link to the Google Cloud Platform Console • Getting a trial account on the Google Cloud Platform sponsored by Google • Getting a trial account on the Google Cloud Platform sponsored by ISB-CGC • General information on how to use the Google Cloud Platform Console • Specific information on how to set up your GCP project for use with ISB-CGC • How to manage project members and change member permissions	• Google Cloud SDK • Google Cloud Shell • Authenticating with Google (two approaches: gcloud init and/or gcloud auth login)
Explore what data are available	• Documentation • GUI	• BigQuery R and Python Tutorials • ISB-CGC Endpoints • Google Genomics APIs
Understand details of the data	• Access controlled data • View File List in GUI • View Sequences with IGV • Google BigQuery Web UI • Google API Explorer	• For data in BigQuery • BigQuery Command Line Tool • BigQuery REST API • For data in Google Cloud Storage • Google Cloud Storage JSON API • Google Cloud Storage gsutil • For data in Google Genomics • Google Genomics REST API
Analyze ISB-CGC data	• Analyses with Bar Charts, Histograms, Scatter Plots, Violin Plots, Cubby Hole Plots and SeqPeek • Google BigQuery Web UI and BigQuery UI Walkthrough	• BigQuery R and Python Tutorials • ISB-CGC Endpoints • Google Genomics APIs
Create cohorts of patients	• Create cohorts • Cohorts set operations • Create a cohort from visualization (background and tool image) • Develop queries with Google BigQuery Web UI and/or Google API Explorer before creating cohorts with command line tools	• Use GUI saved cohorts in ISB-CGC Endpoints • de novo using ISB-CGC Endpoints, BigQuery Command Line Tool, and/or BigQuery REST API
Add your data to the cloud	• Uploading your low level data	• Uploading your data to

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.8 Tutorials and DIY Workshop

The links on this page connect to tutorials and other helpful pages created by the ISB-CGC team:

- DIY Workshop
- Query of the Month Club (BigQuery)
- R, Python and SQL Tutorials
- Feature Alerts
- Moving from GDC to ISB-CGC
- Viewing and using cohorts in the Webapp and API
- Walkthrough of BigQuery
- Using dsub Examples

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.9 Frequently Asked Questions (FAQ)

1.9.1 ISB-CGC Accounts and Cloud Projects

Do I have to request an ISB-CGC account before I can try out the web interface?

No, you can just "sign in" to the web-app using your Google identity.

I want to be able to run big jobs using Google Compute Engine on the TCGA data hosted by the ISB-CGC. What should I do?

You will need to request a Google Cloud Platform (GCP) project. Please see *Your Own GCP project* for more details about requesting a project.

Can I use any email address as a Google identity?

Yes, you can. If your email address is notalready linked to a Google account, you can create a Google account with your current email address. Please note, however that although these two accounts will then share the same *name*, they will still be two separate accounts, with two separate passwords, *etc*. (It is also possible that your institutional email address is *already* a Google account, if your institution uses Google Apps. This is how to find out).

How do I connect my GCP project to the ISB-CGC?

Your GCP project gives you access to all of the technologies that make up the Google Cloud Platform (GCP). These technologies include BigQuery, Cloud Storage, Compute Engine, Google Genomics, etc. The ISB-CGC makes use of a variety of these technologies to provide access to the TCGA data, *without* necessarily inserting an extra interface layer between you and the GCP. Although one component of the ISB-CGC is a web-app (running on Google App Engine), some users may prefer not to go through the web-app to access other components of the ISB-CGC. For example, the open-access TCGA data that we have loaded into BigQuery tables can be accessed directly via the BigQuery web interface or from Python or R. Similarly, the ISB-CGC programmatic API is a REST service that can be used from many different programming languages.

The connection between your GCP project (whether it is an ISB-CGC sponsored and funded project or your own personal project) and the ISB-CGC is your Google identity (also referred to as your "user credentials"). Access to all ISB-CGC hosted data is controlled using access control lists (ACLs) which define the permissions attached to each dataset, bucket, or object.

What project information do I input on the Register a Google Cloud Project page?

You will need to input the Google Cloud Prooject ID which can be found on the Dashboard page of the Google Console under Project info.
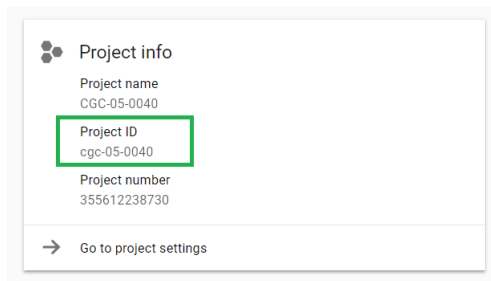
Why do I add the service account 907668440978-oskt05du3ao083cke14641u35deokgjj@developer.gserviceaccount.com to my Google Cloud Project?

This service account is needed in your Google Cloud Project for the ISB-CGC project to be able to automatically verify that all users of your Google Cloud Project have the same appropriate access rights to the protected data that has been reuested for the project.

What service account do I use on the Register a Service Account page to be able to gain access to protected data?

On the Register a Service account page you are asked to input a service account ID. You need to go to the IAM and Admin page which can be found in your console for you Google Cloud Project to find the correct service account. The service account you would like to use is named, "Compute Engine default service account". Please input that service account in the Register a service account page. *Please DO NOT use the service account 907668440978-oskt05du3ao083cke14641u35deokgjj@developer.gserviceaccount.com (you will be prevented from using this account by our software and an error message will be sent indicating this).*

Why can't I re-authorized my Service Account on my Google Cloud Project?

Your service account may have had its permissions revoked (because, for example, the 7-day limit has expired, or you have added a member to the GCP who is not authorized to use that controlled data or has not logged into the ISB-CGC UI and authenticated using their dbGaP credentials). If permissions were revoked because an unauthorized user was added to the project, the Google Cloud Project owner will be sent an email specifying the Service Account, GCP Project, and user which resulted in the access being revoked. If the user has not logged into the ISB-CGC interface and or has not authenticated, you will be given a red error message saying, "There was an error in processing your service account. Please try again." when attempting to refresh using the refresh wheel. To see which new user hasn't logged in or authenticated, please go to the Register a Service Account page and see which user it is within the table for which the dataset is not selected and there are X's in the Registered and Has NIH Identity.


Ensure that the user has 1)Logged into the ISB-CGC user interface and 2) Has registered their NIH Identity with their user interface identity.

To reauthorize the service account 1) remedy the problem that resulted in access being denied, and 2) select the "refresh" icon beside the service account.

Another reason could be if some users are marked as unable to access datasets they should have access to, make sure they have logged into the system and linked their eRA Commons/NIH Identity to their Google Identity.

Why would I get an Authorization Failed page on NIH iTrust when attempting to link my Google Account with my NIH identity?
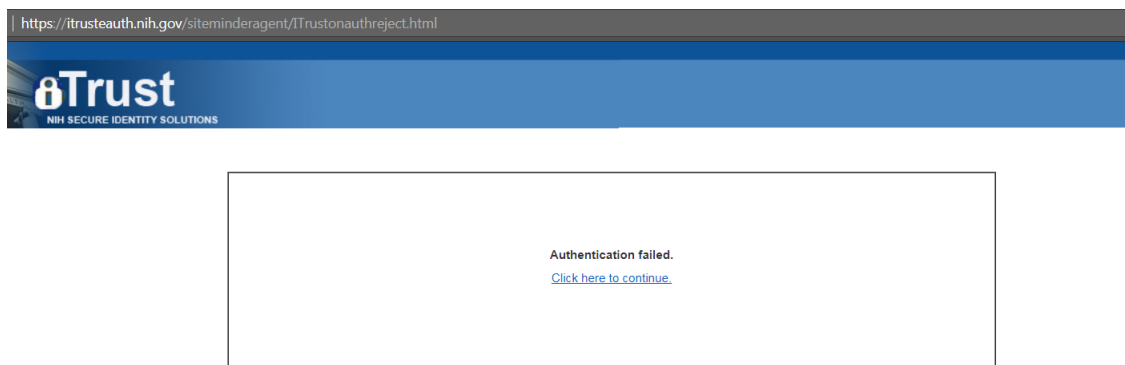
You can get this page for two reasons: First, if you may have typed in your password incorrectly, please select the Click Here to continue link and try to log in again. Second, if you have typed your password correctly, it could time to refresh your NIH identity password. Please reset your password by using this link here and try again. This should allow you to link your NIH Identity to the ISB-CGC user interface.

What happens if I accidently delete the default service account from a Google Cloud Project?

If you accidently delete the default service account associated to the Google Cloud Project you are working in you can no longer authorize the service account during instance creation, associate the service account to controlled access data, and many other functionalities will no longer work.

Then if you try to add the service account back to the Google Cloud Project you return this error,

> *ERROR: (gcloud.compute.instances.create) Some requests did not succeed:*

- The resource 'xx. . . @project.gserviceaccount.com' of type 'serviceAccount' was not found.

Unfortunately at this time, there is no direct way to recover the default service account.

One workaround to recreate the GCE default service account is to disable and re-enable Google Compute Engine API in your project. This will only work if you have no GCE resource (e.g VMs, Disks, Snapshots etc) in your project, otherwise you will get "Backend Provisioning Error" when you try to disable compute engine API.

Another solution would be creating a new project and redeploying your instances there.

Google has an internal feature request to prevent accidental deletion of default service accounts.

There is a Google forum discussion that can be found here with more details and explanation.

1.9.2 ISB-CGC Web Application

Why do I sometimes get a "Do you want to leave this site?" pop-up box when leaving a page or canceling a workflow edit?

This is a security feature when working with forms found in most web browsers; it lets you know that you may have made some changes which will be lost when you navigate away from the page. If you intend to cancel what you were doing, you can safely ignore it.

Which web browser is recomended to use when working with the site?

We recommend using Google Chrome browser. We currently will display a chart slightly off when working with workbooks on a FireFox browser.

Why did I get a 401 error on the IGV Browser?

You will see the 401 error only if your popup blocker is enabled for the isb-cgc website. Please disable the pop-up blocker on the top right-hand side of the screen by selecting to always allow pop-ups from ISB-CGC.

Why does the web browser crash if too many IGV Browser tabs are opened at once?

The web browser will be caused to crash when too many IGV Browser tabs are open due to the memory intensive nature of loading bam files. When working with the IGV Browser please be mindful of having multiple tabs open of the IGV Browser.

Does SeqPeek and CNVR plotting only work with TCGA data?

We currently have no data associated with CNVR or Seqpeek for TARGET or CCLE. Therefore, SeqPeek and CNVR will only work with TCGA data.

1.9.3 Data Access

Does all TCGA data require dbGaP authorization prior to access?

No, generally only the low-level sequence (DNA and RNA) and SNP-array data (CEL files) require dbGaP authorization. All of the "high-level" molecular data, as well as the clinical data are open-access and much of this has been made available in a convenient set of BigQuery tables.

Where can I find the TCGA data that ISB-CGC has made publicly available in BigQuery tables?

The BigQuery web interface can be accessed at bigquery.cloud.google.com. If you have not already added the ISB-CGC datasets to your BigQuery "view", click on the blue arrow next to your project name at the top of the left side-bar, select "Switch to Project", then "Display Project. . . ", and enter "isb-cgc" (without quotes) in the text box labeled "Project ID". All ISB-CGC public BigQuery datasets and tables will now be visible in the left side-bar of the BigQuery web interface. Note that in order to use BigQuery, you need to be a member of a Google Cloud Project.

How can I apply for access to the low-level DNA and RNA sequence data?

In order to access the TCGA controlled-access data, you will need to apply to dbGaP. Please also review our section on **Understanding Data Security**.

I have dbGaP authorization. How do I provide this information to the ISB-CGC platform?

In order for us to verify your dbGaP authorization, you first need to associate your Google identity (used to sign-in to the web-app) with a valid NIH login (*eg* your eRA Commons id). After you have signed in, click on your avatar (next to your name in the upper-right corner) and you will be taken to your account details page where you can verify your dbGaP authorization. You will be redirected to the NIH iTrust login page and after you successfully authenticate you will be brought back to the ISB-CGC web-app. After you successfully authenticate, we will verify that you also have dbGaP authorization for the TCGA controlled-access data. We also ask that you review our section on Understanding Data Security .

My professor has dbGaP authorization. Do I have to have my own authorization too?

Yes, your professor will need to add you as a "data downloader" to his/her dbGaP application so that you have your own dbGaP authorization associated with your own eRA Commons id. (This video explains how an authorized user of controlled-access data can assign a downloader role to someone in his/her institution.)

I already authenticated using my eRA Commons id but now I want to use a different Google identity to access the ISB-CGC web-app. Can I re-authenticate using the same eRA Commons id? Yes, but you will first need to sign-in using your previous Google identity and "unlink" your eRA Commons id from that one before you can link it with your new Google identity. An eRA Commons id cannot be associated with more than one Google identity within the ISB-CGC platform at any one time.

Can I authenticate to NIH programmatically?

No, the current NIH authentication flow requires web-based authentication and must therefore be done from within the ISB-CGC web-app. Once you have authenticated to NIH via the web-app, and your dbGaP authorization has been verified, the Google identity associated with your account will have access to the controlled-data for 24 hours.

1.9.4 Data Content

I get a different number of samples in BigQuery than I do with the same query in the Webapp. Why?

Older programs like TCGA have both legacy data (data from the original program) and harmonized data (data run through the Genomics Data Commons). The Webapp primarily uses harmonized data where BigQuery contains both legacy and harmonized data. In addition, some cases and samples have been removed from the Webapp if annotation suggest the data from those caases or samples are incorrect, misleading or from cases of uncertain origin. Most of these cases and samples are still in BigQuery and users are encouraged to check the annotations tables.

1.9.5 Python Users

I want to write python scripts that access the TCGA data hosted by the ISB-CGC. Do you have some examples that can get me started?

Yes, of course! The best place to start is with our examples-Python repository on github. You can run any of those examples yourself by signing in to your Google Cloud Project and deploying an instance of Google Cloud Datalab.

1.9.6 R and Bioconductor Users

I want to use R and Bioconductor packages to work with the TCGA data. How can I do that?

You can run RStudio locally or deploy a dockerized version on a Google Compute Engine VM. You can find some great examples to get you started in our examples-R repository on github, and also in the documentation from the Google Genomics workshop at BioConductor 2015.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

1.10 Support & Other Useful Links

1.10.1 Contact Us

For general information about the ISB-CGC please contact us at info@isb-cgc.org. We are especially keen on learning about your particular use-cases, and how we can help you take advantage of the latest in cloud-computing technologies to answer your research questions.

For feature-requests or bug-reports, please send e-mail to feedback@isb-cgc.org.

1.10.2 Your Own GCP project

To request an ISB-CGC funded Google Cloud Platform (GCP) project, please send a request to request-gcp@isb-cgc.org. (Note that if you *already* have a GCP project, and are not requesting funds as part of our community evaluation phase, you do not need a separate GCP project in order to work with ISB-CGC hosted data or tools.)

In your request, please describe your research goals in some detail, including information such as the type of data that you plan to use (whether it is your own data that you plan to upload or TCGA data currently hosted by the ISB-CGC), the algorithms and/or methods you plan to apply, and an estimate of the storage and computing costs you expect to incur. Please let us know if you have students or collaborators who will also be accessing the same cloud project. Note that if you are working as a team on a single project, you should all use the same cloud project – if your group is large, we will take this into consideration when determining your funding level.

If you have previous experience using the Google Cloud Platform, that would be useful for us to know – including which specific components (*eg* Compute Engine, BigQuery, Cloud Datalab, *etc*).

All reasonable requests will receive an initial allocation of $300 towards storage and compute costs. We expect that this amount of funding will be more than enough for you to become familiar with the platform. If you expect that you will need additional funding to complete your planned research, this initial amount should be used to perform prototype analyses and to better estimate your total costs. At that time, you may request additional funding.

Please be aware that we will be monitoring your cloud resource usage on a daily basis and will alert you as you begin to approach your funding limit. If you exceed your allocation limit and we are not able to contact you by email for several days, we may need to take action to shut your project down which could cause you to lose work and data.

1.10.3 Other Useful Links

The ISB-CGC platform is built on top of the Google Cloud Platform and has been designed to make the TCGA data as accessible as possible to a wide range of users. For the programmatic users, this includes *complete* access to the tools that Google is pioneering to allow users to scale-up their analyses on the Google infrastructure using a variety of means.

The ISB-CGC documentation and the example code on github will continue to grown to provide starting-points and use-cases designed to suit the needs of a variety of end-users. If you have a particular use-case that has not yet been addressed, please contact us (email info@isb-cgc.org) and we will work with you to determine the best approach to run the analysis you have in mind.

Cloud Datalab is a powerful web-based interactive computational environment built on the familiar IPython (now known as Jupyter) environment, running on a Google VM in your own Google Cloud Project. Cloud Datalab allows you to combine SQL-like queries into the TCGA BigQuery tables with all the power of Python packages like Pandas and Matplotlib. See our examples-Python repository on github.

Google Genomics provides tools for storing, processing, exploring, and sharing DNA sequence reads, reference-based alignments, and variant calls, using Google's infrastructure. An extensive **Cookbook_** here on Read the Docs as well as an ever-growing set of examples on **github_** showcase some of the tools at your disposal.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.

Have feedback or corrections? You can file an issue here or email us at feedback@isb-cgc.org.