Revision History
## CHANGE LOG

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.2</td>
<td>01/23/2015</td>
<td>Fix inaccurate dependency versions, mention 3.8.7</td>
</tr>
<tr>
<td>4.2.1</td>
<td>03/04/2014</td>
<td>Updated Software section for GeneTorrent 3.8.5, added Change Log section</td>
</tr>
</tbody>
</table>
INTRODUCTION

CGHub is a genome sequence storage repository, which provides large capacity and fast access to sequence data generated as part of NCI research projects. The repository is accessed over the Internet, providing simple but secure sequence data search and transfer services. The CGHub currently contains sequence data for The Cancer Genome Atlas (TCGA) cancer research study sponsored by the NCI and NHGRI. In addition the TARGET and CGCI studies’ data is being stored in CGHub.

For these studies, sequence data is considered personally identifying information (PII) because it comes from human clinical samples. In order to support research use guidelines and enforce participant privacy rights, only NIH authorized users of these studies can access the genome sequence data via a protected web services interface (WSI) at CGHub. The metadata that describes the sequence and related analysis, captured as XML-based metadata descriptors, is publicly available via the CGHub WSI.

2.1 Technical Support

For technical or operational requests, or to sign up for the CGHub Announce mailing list please contact: support@cghub.ucsc.edu

2.2 References

CGHub Home Page: https://cghub.ucsc.edu
TCGA Project Home Page at NCI: http://cancergenome.nih.gov
TCGA Data Guide: https://wiki.nci.nih.gov/display/TCGA/TCGA+Data+Primer

2.3 Document Scope

This document describes how to search and download primary genome sequence data and related metadata from the CGHub repository.
Downloading sequence data from CGHub involves three steps listed on the next page:

1. Use your browser to access your eRA account and generate a CGHub authorization credential

2. Query the metadata to generate a list of the desired analysis_id (UUID/s) or create a “manifest.xml” file of multiple analysis_id(s) that contain the sequence data of interest

3. Invoke the GeneTorrent Client with the selected single analysis_id or pass GeneTorrent the list of analysis_id(s) from the manifest.xml to download

These steps will be described in detail in the subsequent sections.

Once a generated authorization credential has been obtained, it can be re-used for an unlimited number of download operations until the credential expires. The credential will need to be periodically renewed (every 365 days or less); a new credential can be obtained at any time. Keep this credential in a secure place on the computer used for downloading. This file should not be emailed to CGHub Support or shared with anyone else.

---

Note: The authorization credential is not required to query the CGHub metadata; the authorization credential is only required for downloading protected data from CGHub. The query functionality is available to the public along with certain non-secured downloads (e.g. the CCLE project data).

Queries can be performed with either the cgquery script or the CGHub Data Browser:

https://browser.cghub.ucsc.edu

- The cgquery script provides a simple but powerful way to execute queries and download files. It is recommended that new users start with the cgquery script before attempting to use the CGHub Web Services API directly.

- The CGHub Web Services API returns an XML result set. This method is useful when scripting or embedding queries into your site-specific tools.

---

1 Some data is publicly available to download using the cghub_public.key credential. Please see Section 5.1 Public Data on how to acquire this “public” key file.

2 If you think your authorization credential has been compromised (e.g. was emailed, put on a public file share, etc…) please download a new credential from https://cghub.ucsc.edu/secure and email us the old credential which we will blacklist so no one gains unauthorized access using it.
This section is specific to the TCGA project and is here to provide background information; it is not required for running downloads from CGHub. The following is in part taken from the TCGA Wiki: https://wiki.nci.nih.gov/display/TCGA/

### 3.1 Organizational Entities

CGHub serves as the central repository for sequencing centers to store genome sequence data, and for analysis centers to retrieve genome sequence data and its associated metadata.

There are four main groups that work together to support the CGHub repository: Tissue Source Sites, Biospecimen Core Resource Centers, Genome Sequencing and Genome Characterization Centers, and Data Coordinating Centers.

A **Tissue Source Site (TSS)** collects samples (tissue, cell or blood) and clinical metadata, which are then sent to a BCR. A TSS is identified by its own TSS ID.

A **Biospecimen Core Resource (BCR)** is a TCGA center where samples are carefully catalogued, processed, quality-checked and stored along with participant clinical information. The work of the BCR includes the following important functions:

- Serving as the interface between the TCGA program and the different Tissue Source Sites that are collecting tumor and matched normal controls
- Ensuring and verifying that TCGA human subjects protections and guidelines are adhered to and that all regulations are followed at each Tissue Source Site
- Examining of all biospecimens to ensure they meet rigorous standards for each tumor type (including percent necrosis and percent tumor nuclei)
- Reviewing of pathology to ensure accurate diagnosis and inclusion in TCGA
- Collecting clinical information for each sample and applying standardized terminology, definitions and formats that are caBIG® compliant
- Extracting and distributing DNA and RNA from samples to each of the genomic characterization and sequencing centers
A **Genome Sequencing Center (GSC)** is a TCGA center that uses high-throughput methods to sequence tumor and normal samples provided by the BCR.

A **Genome Characterization Center (GCC)** is a TCGA center that uses high-throughput technologies to analyze genomic changes involved in cancer. The term “GSC” is used throughout this document to mean either a GSC or a GCC, both submit sequencing data (BAM and FASTQ) files to CGHub and are therefore functionally equivalent for this document’s purposes.

The **Data Coordinating Center (DCC)** is the central provider of TCGA data. The DCC standardizes data formats and validates submitted data.

The work of the DCC includes the following important functions:

- Protecting participant privacy and confidentiality through secure access to research and clinical information that are classified as controlled access datasets
- Developing data standards and controlled vocabularies
- Establishing informatics pipelines for dataflow from production centers to a central repository
- Developing new analytical and visualization technologies for different audiences to facilitate data analysis
- Coordinating project level activities.

### 3.2 System Workflow

The following describes the basic system workflow for creating and distributing genome sequencing data. This workflow is provided here to orient the reader to the end-to-end system operation and is for informational purposes only.

1. **A Tissue Source Site (TSS) submits a participant tissue sample to a Biospecimen Resource Center (BCR) where the sample is stored and prepared for sequencing.**
   - The analyte is shipped to a Genome Sequencing Center (GSC) with a unique identifier (UUID) for the sample Aliquot.
   - Metadata describing the biospecimen is sent to TCGA Data Coordination Center (DCC). The DCC coordinates project level activities and tracks IDs down to the Aliquot level.

2. **GSC receives and processes the analyte from the BCR**
   - The analyte is sequenced by a high throughput sequencer to produce raw read coverage stored in a FASTQ file.
   - Additional processing, such as alignment/mapping is performed and captured in a BAM file.

3. **GSC submits the sequence information to CGHub**
   - A unique identifier (Analysis UUID) for sequence data and metadata information for the single sample analysis is created by the GSC.
   - This set of data (metadata plus sequence and alignment data) is referred to as the Analysis Object.
• GSC performs validation of the metadata prior to upload to ensure proper formatting and the presence of critical metadata elements such as the human assembly reference used for alignments.

• Once validated, the GSC will upload the Analysis Object to CGHub and initiate the workflow on CGHub to make the data available to downloaders. Currently the Analysis Object can either be a BAM (sequence + alignments) or a FASTQ file (sequence only).

4. CGHub synchronizes with the TCGA DCC to determine status of sample.

• The DCC maintains the higher-level study attributes as well as clinical data associated with each sample (however clinical data is not stored at CGHub).

• CGHub will query the DCC to verify that submitted data is associated with a valid sample that should be available to authorized researchers.

• CGHub will also query for redacted samples and suppress the sequence data from download.

5. Researchers retrieve the CGHub catalog and perform queries across the public sequence and alignment metadata using selected metadata attributes such as cancer type, sequence type, source sequencing center, or date range among others.

6. Researchers retrieve protected sequence and alignment data using their NCI authorized credentials, as long as the sequence data is in the live state.

3.3 Related Documents

• CGHub Home Page: https://cghub.ucsc.edu/

• TCGA Project Home Page at NCI: http://cancergenome.nih.gov/

• TCGA Data Guide: https://wiki.nci.nih.gov/display/TCGA/TCGA+Data+Primer


4.1 cgquery - The Metadata Query Utility

cgquery is a command line tool used to search the metadata contained within CGHub. Once you have queried data that you are interested in downloading, you may use the analysis_id (UUID) listed from the cgquery output, and pass a single analysis_id to GeneTorrent, or you can create a manifest.xml file with multiple analysis_id’s in it, and pass this file to GeneTorrent to download several analysis_id’s iteratively. Please see section 6 for cgquery examples.

4.1.1 Prerequisites

Python 2.6.5 or later with SSL support is required for Linux and Mac users.

1. Check for Python Version

   $ python -c 'import platform; print platform.python_version()'

   Note: Version 2.6.5 through 2.7.2 is required

2. Check if Python supports SSL

   Use the following command to check if the python installation supports SSL:

   $ python -c 'import socket; print hasattr(socket, "ssl")'

   Note: True is required

4.1.2 Installation

cgquery is included with the GeneTorrent client package for all supported operating systems.

4.2 gtdownload - The Data Download Program

gtdownload is also known as GeneTorrent Client Program.
4.2.1 Supported Platforms

Binary (ready-to-run) versions of the client are available for 64-bit versions of the following operating systems:

**LINUX Systems**

- CentOS 5 & 6 (may also work with corresponding Redhat release)
- Ubuntu 10.04, 12.04 and 13.10
- Fedora 19 & 20

**Macintosh Systems**

- Mac OS X Lion 10.7 (64-bit) (not officially supported)
- Mac OS X Mountain Lion 10.8 (64-bit)
- Mac OS X Mavericks 10.9 (64-bit)

**Windows Systems**

- Windows XP (32 bit) (not officially supported)
- Windows Vista (32/64 bit) (not officially supported)
- Windows 7 (32/64 bit)
- Windows 8 (32/64 bit)
- Windows Server 2008 (64 bit)

There are plans to expand this list in future, please check the software downloads page for updates:

https://www.cghub.ucsc.edu/software/downloads.html

The source code for GeneTorrent can also be found at the same location as above.

4.2.2 Prerequisites

**Linux Systems**

The GeneTorrent software requires certain software packages to be installed to function on Linux systems. For the cases where the dependency is listed as Required this usually means that in that platform/OS’s case the dependency is probably already installed as part of a typical OS install or is accessible through the normal package update methods employed by CentOS/RHEL/Fedora (yum) or Ubuntu (apt-get). The table below lists the dependencies for GeneTorrent version 3.8.7.
### Table 4.1: GeneTorrent dependencies of 3.8.7+ binary .tar.gz packages

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>boost</td>
<td>Bundled 1.48.0</td>
<td>Bundled 1.48.0</td>
<td>Bundled 1.48.0</td>
<td>Bundled 1.48.0</td>
<td>Required 1.0.1e</td>
<td>Required 1.0.1e</td>
<td>Required 1.0.1e</td>
<td>Required 1.0.1f</td>
<td>Bundled² 1.50.0</td>
<td>Bundled 1.48.0</td>
</tr>
<tr>
<td></td>
<td>openssl</td>
<td>Bundled 1.0.1g</td>
<td>Required 1.0.1e</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 1.0.1g</td>
<td>Provided 0.9.8³</td>
</tr>
<tr>
<td></td>
<td>xerces-c</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.1.1</td>
<td>Bundled 3.0.1</td>
<td>Bundled 3.1.1</td>
</tr>
<tr>
<td></td>
<td>xqilla</td>
<td>Bundled 2.3.0</td>
<td>Bundled 2.3.0</td>
<td>Bundled 2.3.0</td>
<td>Bundled 2.3.0</td>
<td>Bundled 2.3.0</td>
<td>Bundled 2.3.0</td>
<td>Bundled 2.3.0</td>
<td>Bundled 2.3.0</td>
<td>Bundled 2.3.0</td>
<td>Bundled 2.3.0</td>
</tr>
<tr>
<td></td>
<td>libcurl¹</td>
<td>Required 7.15.5-15</td>
<td>Required 7.19.7-37</td>
<td>Required 7.29.0</td>
<td>Required 7.32.0</td>
<td>Required 7.19.7</td>
<td>Required 7.22.0</td>
<td>Required 7.32.0</td>
<td>Required 7.35.0</td>
<td>Required 7.34.0</td>
<td>Provided 7.37.1³</td>
</tr>
</tbody>
</table>

**Bundled:** GeneTorrent comes with these packages bundled in with the GeneTorrent client. If an older version of a dependency is already installed on the system, the version bundled with GeneTorrent will coexist peacefully since it is installed in a separate location on the system.

**Required:** The listed version of the package will need to be installed. The version of the dependency is the same as the one provided by the operating system’s native package manager. Installing this package typically requires administrator privileges.

**Provided:** This package is provided by a default install of the operating system.

---

¹ GeneTorrent bundles this dependency from Cygwin, rather than creating it at build time.
² This dependency is provided out of the box on OS X.
³ The `libcurl` package is called `curl` on CentOS 5 and `libcurl3` on Ubuntu.

In versions before 3.8.7, if a native GeneTorrent package (.deb and .rpm) bundles a dependency, so does the binary .tar.gz package. A native package of any GeneTorrent version only bundles these three dependencies if the target Linux distribution does not already provide a suitable version of that dependency as a native package. Starting with version 3.8.7, the binary .tar.gz package always bundles Boost 1.48.0, Xerces-C 3.1.1 and Xqilla 2.3.0 on every Linux distribution even if the target distribution provides a suitable native package. This change was made in order to decrease the odds of having to involve an administrator when installing GeneTorrent from the binary package.
Table 4.2: GeneTorrent dependencies of native 3.8.7+ packages and all 3.8.6 packages

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Dependency</th>
<th>Operating System</th>
<th>Dependency</th>
<th>Operating System</th>
<th>Dependency</th>
<th>Operating System</th>
<th>Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>boost</td>
<td>openssl</td>
<td>xerces-c</td>
<td>xqilla</td>
<td>libcurl1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CentOS 5/RHEL</td>
<td>Bundled</td>
<td>Bundled</td>
<td>Bundled</td>
<td>Bundled</td>
<td>Required</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.48.0</td>
<td>1.0.1g</td>
<td>3.1.1</td>
<td>2.3.0</td>
<td>7.15.5-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CentOS 6/RHEL</td>
<td>Bundled</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.48.0</td>
<td>1.0.1e</td>
<td>3.0.1</td>
<td>2.2.3</td>
<td>7.19.7-37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fedora 19</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.53.0</td>
<td>1.0.1e</td>
<td>3.1.1</td>
<td>2.2.4</td>
<td>7.29.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fedora 20</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.54.0</td>
<td>1.0.1e</td>
<td>3.1.1</td>
<td>2.3.0</td>
<td>7.32.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 10.04 LTS</td>
<td>Bundled</td>
<td>Required</td>
<td>Required</td>
<td>Bundled</td>
<td>Required</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lucid</td>
<td>1.48.0</td>
<td>0.9.8k</td>
<td>3.1.1</td>
<td>2.3.0</td>
<td>7.19.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 12.04 LTS</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precise</td>
<td>1.48.0</td>
<td>1.0.1</td>
<td>3.1.1</td>
<td>2.3.0</td>
<td>7.22.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 13.10 Saucy</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.53.0</td>
<td>1.0.1e</td>
<td>3.1.1</td>
<td>2.3.0</td>
<td>7.32.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 14.04 LTS</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trusty</td>
<td>1.54.0</td>
<td>1.0.1f</td>
<td>3.1.1</td>
<td>2.3.0</td>
<td>7.35.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>Bundled2</td>
<td>Bundled2</td>
<td>Bundled2</td>
<td>Bundled</td>
<td>Bundled2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.50.0</td>
<td>1.0.1g</td>
<td>3.0.1</td>
<td>2.3.0</td>
<td>7.34.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac OS X</td>
<td>Bundled</td>
<td>Provided</td>
<td>Bundled</td>
<td>Bundled</td>
<td>Provided</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.48.0</td>
<td>0.9.8</td>
<td>3.1.1</td>
<td>2.3.0</td>
<td>7.37.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Macintosh Systems

No prerequisites are required.

Windows Systems

No prerequisites are required.

4.2.3 Download

For all versions of gtdownload (GeneTorrent):

To find the GeneTorrent software installation package, navigate to the GeneTorrent Software Downloads page: https://cghub.ucsc.edu/software/downloads.html

Linux Systems

For installing GeneTorrent on one of the supported Linux distributions you have the choice of either a binary tar.gz package or a set of packages for installation via the distribution’s native package manager. The tar.gz
package can be installed without root privileges to an arbitrary location on the target system provided that the necessary dependencies are already installed.

The tar.gz packages are named `gtdownload-x.x.x-n-*YOUR_OS*.x86_64.tar.gz`, where `x.x.x` is the current version of GeneTorrent such as `3.8.5`, `n` is the build number and `*YOUR_OS*` is the name of the target Linux distribution.

The naming of the native .deb and .rpm files follows the conventions of the respective package manager.

### 4.2.4 Installation

#### Linux Systems

**Native packages (.rpm or .deb)**

.rpm packages of GeneTorrent are available for Linux distributions using the RPM package manager. You will need to download two packages: the `GeneTorrent-common-*.*.rpm` and `GeneTorrent-download-*.*.rpm`. On CentOS 5, for example, the packages can be installed using `yum` install:

```
sudo yum --nogpg localinstall \
GeneTorrent-common-3.8.5-11.91.el5.x86_64.rpm \
GeneTorrent-download-3.8.5-11.91.el5.x86_64.rpm \
```

Be sure to download and install the .rpm files for your particular distribution. The .el5. designator in the above example identifies the package as CentOS 5. For Centos 6, Fedora 19 or Fedora 20 the designator would read .el6., .fc19. or .fc20., respectively.

CGHub also provides .deb packages for Ubuntu. You will need to download two packages: `genetorrent-common_*.*.deb` and `genetorrent-download_*.*.deb`. On Ubuntu 12.04 “Precise”, for example, you would install the packages using the following commands:

```
sudo dpkg -i \
genetorrent-common_3.8.5-ubuntu2.91-12.04_amd64.deb \
genetorrent-common_3.8.5-ubuntu2.91-12.04_amd64.deb
```

If any of GeneTorrent’s dependencies are missing, dpkg will display warning messages and leave the packages unconfigured. To fix this, simply run:

```
sudo apt-get --fix-broken
```

**Binary .tar.gz package**

If you need to install GeneTorrent as a non-privileged user, please download the .tar.gz package of GeneTorrent (available from the same location as the packages above):

```
cd
 tar -xzvf GeneTorrent-download-3.8.5-91-CentOS6.4.x86_64.tar.gz
```
This will create a cghub directory in your home directory. Depending on your target Linux distribution, you might have to ask your administrator to install additional dependencies according to the dependency_matrix above. Consider adding $HOME/cghub/bin to your PATH.

Macintosh Systems

Be sure to check Allow applications downloaded from: Anywhere under System Preferences -> Security & Privacy

- Click on the GeneTorrent-3.8.5-23.pkg you downloaded from CGHub’s website (https://cghub.ucsc.edu/software/downloads.html)
- This will open the GeneTorrent Installer.
- Click Continue.
- Read and agree to the user’s agreement.
- Select the destination to where the gtdownload program will be installed (the default location is preferred) and click Install.
- Provide the Administrator’s password for installing programs on your system.
- At this point, the installation will begin to take place, click closed when finished.

Windows Systems

- As an administrator, double click on the GeneTorrent-3.8.5-45-installer.exe you downloaded from CGHub’s website.
- This will open a User Account Control window to let you know that you are installing a program on your system, please click Yes to proceed.
- The GeneTorrent License Agreement window will appear and you will need to check the I accept the terms... check box and click Next.
- Default installation will be selected, and click Next
- Select the destination folder (default location is preferred), and click Install.

Note: Some Windows users may not be able to open the GeneTorrent Shell command prompt (cmd) shortcut on their desktop. Please contact your local IT support team, and be sure that you have permissions to open a “cmd” window. Alternatively, you may be able to open a “cmd” window but not the shortcut, in which case navigate to the GeneTorrent folder and run the gtdownload or cgquery commands. Example: C:\>cd "c:\Program Files (x86)\GeneTorrent” and run gtdownload or cgquery from the GeneTorrent Folder.
4.2.5 Verifying the Installation

To verify that the installation succeeded and that the correct version of the software is installed type the following command in a terminal/console window:

**Linux and OS X**

Run:

```
gtdownload --version
```

**Windows**

1. Open **Start -> Programs -> GeneTorrent -> GeneTorrent Shell**
2. Type `gtdownload --version`

The expected result should be similar to the below output for Linux, Mac and Windows systems:

```
GeneTorrent gtdownload release 3.8.5 (SCM REV: git ref: 1b8e7685d6, build: 91)
```

For technical or operational requests, contact support@cghub.ucsc.edu.
NETWORK/FIREWALL CONSIDERATIONS

If you are on a network that is managed by a security team, you may need to contact the network administrators to open ports or add exceptions on your system or your firewalls.

Please open up outbound, TCP access to the following IP address block with a class C subnet: 192.35.223.0/24

If your network location is running an Intrusion Prevention System (IPS), you may need to allow an exception for BitTorrent tracker traffic to: 192.35.223.52:21111

CGHub clients require the following firewall configuration:

- Open up the firewall for only outgoing TCP connections to CGHub. Incoming connections are not required.
- We recommend allowing access to the full CGHub network:

<table>
<thead>
<tr>
<th>Host</th>
<th>IP Address</th>
<th>Protocol</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>*.cghub.ucsc.edu</td>
<td>192.35.223.0/24</td>
<td>TCP</td>
<td>443, 20893-20923, 21111</td>
</tr>
</tbody>
</table>

This provides a simple configuration and prevents problems as CGHub adds more servers to accommodate growing capacity. It is possible to use a more restrictive IP address range.

- If the number of systems accessing CGHub is limited, configure firewalls to allow traffic to CGHub from only those systems.
- White list the CGHub servers to not perform intrusion prevention deep packet inspection (DPI), as it will slow down transfers and is not useful for SSL encrypted connections.

For more information, for you or your network administrators, regarding security and running GeneTorrent, please refer to the CGHub Client Security Guide available on our help page here:
6.1 Public Data

CGHub hosts public data (e.g. CCLE cancer cell lines), and protected data (e.g. TCGA, TARGET, CGCI). The public data is available for anyone to download.

You can do this by passing GeneTorrent the URL to the key file as in this example:

```
$ gtdownload -vv -d 89324e86-5b7a-4f69-92c1-3b67293f8748 -c https://cghub.ucsc.edu/software/downloads/cghub_public.key
```

You can do this in interactive mode as well with cgquery using the -i (interactive) option:

```
cgquery -i "study=*Multi*" -c https://cghub.ucsc.edu/software/downloads/cghub_public.key
```

6.2 Secure Data

**In order to download secure CGHub data:**

1. Users must first have a valid eRA Commons account
2. Must have received authorization from the Data Access Committee (DAC) for the appropriate project:

Table 6.1: Table of possible values for the study name for secure data sets.

<table>
<thead>
<tr>
<th>Project Abbreviation</th>
<th>Study ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCGA</td>
<td>phs000178</td>
</tr>
<tr>
<td>Pan-Can (subset of TCGA)</td>
<td>PAWG</td>
</tr>
<tr>
<td>TARGET</td>
<td>phs000218</td>
</tr>
</tbody>
</table>

Please refer to the CGHub New User’s authentication page for more information on applying for access to controlled-access data: https://www.cghub.ucsc.edu/keyfile/newuser.html
6.2.1 Authentication

Steps 1 - 4 are only required to generate a new key. Download commands can be run repeatedly using the same cghub.key file until it expires (after 365 days). Please visit our website: https://cghub.ucsc.edu/keyfile/newuser.html for more information for new users.

1. Access URL for CGHub Authentication page.
   The user begins to log in at the URL: https://cghub.ucsc.edu/secure.

2. The CGHub website redirects the user to the NIH login page, shown below. The user provides valid eRA login credentials (Username and Password).

   ![Login Page](image)

3. NIH authenticates the user.

4. CGHub generates and returns an authorization credential called cghub.key.

   **Attention:** Please keep this file secured, as it is sensitive and should never be transmitted over unsecured media such as email.

The user will be prompted to save the authorization credential in a local file. This credential is valid for **365 days**. You will need to repeat this process to generate a new credential whenever the credential expires, the user’s account changes, or the credential file is compromised.

**Note:** Users coming from within NIH

Everyone including NIH users with other NIH accounts (NIH/CIT) MUST have an eRA account to access CGHub. Your NIH/CIT/other login will not work even if you were granted access by the DAC for that login. We strongly encourage NIH users to obtain an eRA account BEFORE applying for the DAC authorization. Then when you apply you will use your eRA account login as the one you’re requesting authorization from the DAC for. You may also need to contact your local NIH IC Coordinator to update your certificates to work with the **eRA Commons** account. Please visit the NIH IC Coordinators website to find your local IC Coordinator: http://grants.nih.gov/grants/policy/coc/contacts.htm

6.2. Secure Data
The authorization (from the Data Access Committee/dbGaP) and credential file (cghub.key) is usually a different expiration date depending on when you downloaded the cghub.key file. Both must be kept up-to-date for you to download from CGHub. The best way to do this currently is to simply download a credential file (cghub.key) from CGHub at the same time you receive authorization/re-authorized notice from dbGaP. Both (authorization and credential file) must be renewed/downloaded every 365 days.

Pass the authorization credential file to the gtdownload client as part of the sequence data download step with the –c option.
CHAPTER
SEVEN

SEQUENCE METADATA QUERIES

7.1 Identifiers

Identifiers in the TCGA project are assigned at multiple levels of the data. Universally unique identifiers (UUIDs) are assigned for the Participant (patient), the Sample, the Aliquot, and the Analysis which acts as the primary ID at CGHub. Analysis IDs at CGHub are handled the same across all projects (not just TCGA), though the higher level ID structure may change.

Aliquots are the most specific level of the data directly tracked at the TCGA DCC (please see the System Overview section) and are one level above the CGHub controlled Analysis id. A TCGA Aliquot UUID is equivalent to the TCGA barcode, e.g. TCGA-CS-6188-01A-11R-1896-07. The Aliquot UUID was introduced to be used in addition to the TCGA barcode to avoid problem updating the embedded metadata present in the TCGA barcode (such as the analyte id, 11R in the example).

The Analysis ID at CGHub was introduced due to the fact that one Aliquot ID can have multiple data files (e.g. BAM or FASTQ) associated because of differences in the human assembly reference used in the alignment (e.g. HG18 vs. HG19), differences in the alignment parameters themselves, or other changes which produce multiple data files from the same starting source material. Thus the Analysis ID at CGHub is uniquely tied to one and only one specific data file throughout the life of the ID. If anything in the underlying data file changes a new Analysis ID will be assigned and a new metadata record will be created at CGHub. However, the old Analysis id and metadata record will continue to exist for historical purposes.

7.2 Basic Organization

Each data file (e.g. BAM or FASTQ) in CGHub is described by metadata, which contains attributes such as the date, sample type, and disease abbreviation. (See section 6.1 for the list of searchable attributes). Querying the CGHub metadata is a powerful method of identifying data files of interest. The cgquery program is a command line tool that allows the user to construct and submit a query string to search for data files with one or more attributes. The results are formatted in either a summary or complete view. (See section 6.2 cgquery)

cgquery uses the public CGHub REST APIs. Users can call this interface directly through a browser or other tool such as wget or curl and develop custom GUIs or command line tools. (See section 6.4 for more information about the REST APIs)
Whether using cgquery, the CGHub Data Browser, or the REST APIs directly, the attributes that make up the query are the same (please see the next subsection including the tables for details)

Most sequencing centers submit only their BAM data files to CGHub. However, in the case of the centers which submit RNA-seq data, the FASTQs are/will be uploaded as well (e.g. University of North Carolina [UNC-LCCC]). This allows the user to have the option of downloading just the sequences and base qualities if they don’t want the alignments for these RNA-seq experiments. Thus the term “data files”, used throughout this document, refers to both BAM and FASTQ files stored at CGHub.

Note: FASTQ files are usually split into at least 2 files (2 sets of read mates) which are then compressed (usually with a gzip of a TAR archive) at CGHub and downloaded in that form.

7.3 Query Attributes

The query string can include one or more of the attributes listed below using attribute_name=attribute_value pairs.

Multiple attribute_name queries can be combined using the & symbol, and multiple attribute_value queries can be combined using the OR in the description.

For example:

cgquery "disease_abbr=OV&sample_type=01&library_strategy=(WGS OR RNA-Seq)"

Both attribute names and attribute values are case sensitive.

Values can use the * wildcard. For example, disease_abbr=O* will match any values beginning with the letter O.

The “-” in front of the attribute name will excluded these queries. For example the following will not include sample_type=01:

cgquery "disease_abbr=OV-&sample_type=01&library_strategy=(WGS OR RNA-Seq)".

CGHub also stores the original submission XML documents, which users can perform free text searches against any strings within the analysis.xml, run.xml, and experiment.xml files. These xml_text queries are case sensitive.

For more information about constructing query strings see the Solr Query Syntax Wiki and the Lucene Query Documentation.

A list of supported query attributes can be found in the following sections.

7.3.1 CGHub Specific Metadata Fields

**analysis_id**

is the primary Globally unique ID (UUID) associated with a collection of sequencing data files and their metadata. For example:

analysis_id=1baaf59d-f9d6-4f78-ad96-c3e9089de8bb
Generated by GSC prior to submission. Legacy data was assigned uuids when it was migrated from SRA into CGHub.

**state**

determines if the submission is still being processed (uploading), live, redacted, or suppressed (among others). Users can only download data that is in the “live” state. For a complete list of “states”, please see the [Submitters Guide Section 3.1](#). States are managed by CGHub. For example:

```
state=live
```

**last_modified**

is the date the object was last modified.

For example, to find all submissions modified in the last six months:

```
last_modified=[NOW-6MONTH TO NOW]
```

For example, to find all submissions modified between July 16 and July 25, 2011:

```
```

Created by CGHub.

**study**

The study for which the data was generated. This value is used to determine which users can download the data. For example:

```
study=phs000178
```

Is the TCGA study reference and is a “secure data set” (as denoted below) and you must have explicit authorization from dbGaP for all secure data sets.

**Secured Data Sets:**

1. The Cancer Genome Atlas (TCGA) phs000178

2. Pan-Cancer Analysis of Whole Genomes PAWG project study name.

3. Therapeutically Applicable Research to Generate Effective Treatments (TARGET) phs000218 (TARGET’s parent study number for which you request access to)

   - phs000463 (Child study inherited access from phs000218: Acute Lymphoblastic Leukemia (ALL) Pilot Phase 1)
   - phs000464 (Child study inherited access from phs000218: Acute Lymphoblastic Leukemia (ALL) Expansion Phase 2)
   - phs000465 (Child study inherited access from phs000218: Acute Myeloid Leukemia [AML])
   - phs000466 (Child study inherited access from phs000218: Kidney, Clear Cell Sarcoma of the Kidney [CCSK])
   - phs000467 (Child study inherited access from phs000218: Neuroblastoma [NBL])
   - phs000468 (Child study inherited access from phs000218: Osteosarcoma [OS])

### 7.3. Query Attributes
• phs000469 (Child study inherited access from phs000218: Model Systems (MDLS): Cell Lines and Xenografts [including PPTP])
  • phs000470 (Child study inherited access from phs000218: Kidney, Rhabdoid Tumor [RT])
  • phs000471 (Child study inherited access from phs000218: Kidney, Wilms Tumor [WT])

To query all TARGET studies use this example:

study=(phs000218 OR phs0004*)

3. Cancer Genome Characterization Initiative (CGCI) phs000235

Public Data Sets:

1. Cancer Cell Line Encyclopedia (CCLE) Homo sapiens Other_Sequencing_Multiisolate which could be shortened to "*multi*" utilizing the wildcard "*".

CCLE example query:

study=*multi*

2. Benchmark 4 Data TCGA_MUT_BENCHMARK_4

The study data field is extracted from analysis.xml <STUDY_REF refname=>

xml_text
Free form search from the original submission XML documents. For example:

xml_text=NOMINAL_LENGTH

Covers the content of analysis.xml, run.xml and experiment.xml.

7.3.2 Aliquot and Sample Metadata Definitions

aliquot_id
Globally unique ID for the aliquot used to generate this analysis. This is a primary identifier for each study’s sample database (e.g. TCGA’s DCC Portal). Example:

aliquot_id=ff9e47fa-c8b6-4005-b0ba-c9abf59752ca

Extracted from analysis.xml.

disease_abbr
The short disease name. The set of legal TCGA values are defined by the DCC codes table at: http://tcga-data.nci.nih.gov/datareports/codeTablesReport.htm. TARGET uses similar but slightly different conventions. Most prominently, TARGET does not define abbreviations for all diseases it covers. Where possible, an abbreviation consistent with TCGA is returned for TARGET samples, otherwise the full name of the disease is used.

disease_abbr=OV

Imported from the TCGA DCC ¹ or the TARGET DCC ².

---

¹ https://tcga-data.nci.nih.gov/uuid/uuidws/metadata/xml/uuid/<aliquot_uuid>
² http://ocg.cancer.gov/programs/target

7.3. Query Attributes
**participant_id**

The ID of the human participant providing the sample. Previously represented by the TCGA 3 part barcode (e.g. TCGA-AB-2802). For TCGA this will be a participant UUID provided by the DCC. TARGET values follow the TARGET barcode ID format.

TCGA: participant_id=aeaf19a-3b0d-4c79-a029-44642581f4d8

TARGET: participant_id=TARGET-30-PALHVD

Imported from the TCGA DCC ¹ or the TARGET DCC ².

**sample_id**

The ID for the root sample from which the aliquot was derived. Previously represented by the TCGA 4 part barcode (e.g. TCGA-AB-2802-03C). For TCGA this will be a sample UUID provided by the DCC. TARGET values follow the TARGET barcode ID format.

TCGA: sample_id=972a7b73-9323-4b02-90dd-bb420a18681f

TARGET: sample_id=TARGET-30-PALTYB-10A

Imported from the TCGA DCC ¹ or the TARGET DCC ².

**analyte_code**

The type of analyte (e.g. D=DNA, R=RNA, etc.). The set of legal TCGA values are defined by the DCC at: http://tcga-data.nci.nih.gov/datareports/codeTablesReport.htm. TARGET values are inferred from the sample barcode ID.

TCGA: analyte_code=D (for non-amplified DNA)

TARGET: analyte_code=W (for amplified DNA), using TARGET-30-PALTYB-10A-01W

Imported from the TCGA DCC ¹ or the TARGET DCC ².

**sample_type**

The type of sample (e.g. sample_type=01 Blood Derived Normal, Primary solid Tumor, etc.). The set of legal TCGA values are defined by the DCC codes table at: http://tcga-data.nci.nih.gov/datareports/codeTablesReport.htm. TARGET values are inferred from the sample barcode ID and prefixed with the string “TARGET: “. The prefix is precautionary measure to avoid confusion with TCGA sample types which use a similar but different convention.

TCGA: sample_type=01

TARGET: sample_type=10, using TARGET-30-PALTYB-10A-01W

Imported from the TCGA DCC ¹ or the TARGET DCC ².

**tss_id**

The Tissue Source Site. The set of legal TCGA values are defined by the DCC codes table at: http://tcga-data.nci.nih.gov/datareports/codeTablesReport.htm. Not available for TARGET. For example:

```
tss_id=16
```

Imported from the TCGA DCC ¹.

**sample_accession**

Legacy sample accession # for data migrated from SRA. For example:
sample_accession=SRS194938

Extracted from `<TARGET accession=...> in analysis.xml`

**analysis_accession**
Legacy analysis accession number from SRA. Of the form SRZ#### where each # stands for a single digit. For example:

    analysis_accession=SRZ014781

Extracted from `<ANALYSIS accession=...> in analysis.xml`.

### 7.3.3 Additional Metadata Field Definitions

**center_name**
Short names defined by the project. Please reference the Center Name Table below. For example:

    center_name=BI

Extracted from `<ANALYSIS center_name=...> in analysis.xml`.

**refassem_short_name**
This is the reference assembly short name. Below are the available searchable values.

- GRCh37
- GRCh37_BI_Variant
- GRCh37-lite
- GRCh37-lite+-HPV_Redux-build
- HG18
- HG18_Broad_variant
- HG19
- HG19_Broad_variant
- NCBI36_BCCAGSC_variant
- NCBI36_BCM_variant
- NCBI36_WUGSC_variant
- NCBI Data file extensions: build36
- unaligned (FASTQ)

For example:

    refassem_short_name=HG18

Note: the unaligned reference is for data files which do not contain alignments, such as FASTA/FASTQ files.
Please see the CGHub Data Browser help pages for more information on reference assemblies: https://browser.cghub.ucsc.edu/help/assemblies/

Extracted from `<ANALYSIS short_name=>` in analysis.xml.

**filename**
Name of a data file associated with an analysis_id. filename=C282.TCGA-06-2558*

Extracted from `<FILES>` element in analysis.xml and/or experiment.xml. Data file extensions consist usually of either .bam or .tar.gz (for FASTQs).

Note: filenames are not controlled or validated by CGHub, this is per official TCGA practice where the associated IDs (e.g. analysis_id) are the primary identifiers and are used to uniquely track metadata and data entries.

**filetype**
Either bam or fasta

(fasta is used for FASTQ files due to a limitation in the underlying SRA metadata schema).

filetype=bam

filetype=fasta

Extracted from `<FILES>` element in analysis.xml.

**filesize**
Used to search for data files within a size range in bytes. For example:

filesize=[80530636800 TO *]

Determined by CGHub as the files are submitted. (80530636800 Bytes = ~75 Gigabytes)

**analysis_type**
The type of analysis. Uses the values from SRA 1.5 schema. For TCGA, this will be REFERENCE_ALIGNMENT:

analysis_type=REFERENCE_ALIGNMENT

Extracted from `<ANALYSIS_TYPE>` in analysis.xml.

**library_strategy**
The sequencing technique used. Values are defined by the 1.5 SRA schema: Commonly used library strategies in the CGHub repository:

•VALIDATION
•RNA-Seq
•miRNA-Seq
•WGS
•WXS
•Bisulfite-Seq
•OTHER

Library strategies not commonly found in the CGHub repository but still supported by CGHub:

7.3. Query Attributes
• AMPLICON
• ChIP-Seq
• CLONE
• CLONEEND
• DNAse
• EST
• FINISHING
• FL-cDNA
• CTS
• DNase-Hypersensitivity
• MDB-Seq
• MeDIP-Seq
• miSRA-Seq
• MNase-Seq
• MRE-Seq
• POOLCLONE
• WCS

Example:

    library_strategy=WGS

Extracted from `<LIBRARY_STRATEGY>` in experiment.xml.

**platform**

The sequencing machine used to generate the sequence data.

Values are defined by the SRA 1.5 schema:

• LS454
• ILLUMINA HELICOS ABI_SOLID
• COMPLETE_GENOMICS PACBIO_SMRT
• ION_TORRENT

Example:

    platform=ILLUMINA

Extracted from `<PLATFORM>` in experiment.xml.

**alias**

Name of the analysis object as defined by the submitting center. For example:

    alias=m03270

---

7.3. Query Attributes

27
Extracted from <ANALYSIS alias=> in analysis.xml.

title
Title of the analysis as defined by the submitting center.
Extracted from <TITLE> in analysis.xml.

Table 7.1: Table of possible values for the center_name column

<table>
<thead>
<tr>
<th>Center</th>
<th>center_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC Cancer Agency Michael Smith Genome Sciences Center</td>
<td>BCCAGSC</td>
</tr>
<tr>
<td>Baylor College of Medicine</td>
<td>BCM</td>
</tr>
<tr>
<td>Broad Institute</td>
<td>BI</td>
</tr>
<tr>
<td>Complete Genomics Incorporated</td>
<td>CompleteGenomics</td>
</tr>
<tr>
<td>Harvard Medical School - Raju Kucherlapati Lab</td>
<td>HMS-RK</td>
</tr>
<tr>
<td>University of California Santa Cruz</td>
<td>UCSC</td>
</tr>
<tr>
<td>University of North Carolina at Chapel Hill – Lineberger Comprehensive Cancer Center</td>
<td>UNC-LCCC</td>
</tr>
<tr>
<td>University of Southern California - Peter Laird’s group</td>
<td>USC-JHU</td>
</tr>
<tr>
<td>Washington University, Genome Sequencing Center</td>
<td>WUGSC</td>
</tr>
</tbody>
</table>

7.4 Using cgquery

The cgquery script calls the Web services APIs and parses the XML output into a human readable form. It takes the same query string arguments as the direct Web Services API.

Usage: cgquery [options] <querystring>

-version
show program’s version number and exit

-h, -help
show this help message and exit

-o OUTPUTXML, -outputxml=OUTPUTXML
save output to a file as raw xml instead of displaying formatted results

-s SERVER, -server=SERVER
CGHub server location, including protocol and port, e.g. https://cghub.ucsc.edu

-S SORTBY, -sort=SORTBY
attribute by which results should be sorted. May be followed by an :asc or :desc to indicate ascending or descending order.

-g GTBIN, -gt-binary=GTBIN
the GeneTorrent binary used in interactive mode. Use a fully qualified path if the binary is not in your PATH. e.g. /usr/bin/GeneTorrent
-I, -ident
query only the minimal set of fields to identify the analysis object

-b SUBMISSION, -submission=SUBMISSION
query only the named submission metadata, or 'all' for all metadata.

-f CONFDIR, -conf-dir=CONFDIR
path to the GeneTorrent configuration file.

-a, -attributes
query the full set of analysis attributes, including submission metadata.

-i, -interactive
enable interactive mode

-n, -count-only (default)
report only a count of matching results. Not available in interactive mode.

-A, -all-states
output non-downloadable objects. By default only downloadable objects are displayed or written to
the XML output file.

-c CREDENTIAL, -credential=CREDENTIAL
file containing the GeneTorrent credential. Only required for interactive mode.

-v or -vv, -verbose
enable verbose output, even when writing XML output to a file.

<querystring>
should be the fully quoted query string, without the question mark separator, e.g.
disease_abbr=COAD.

7.5 Example cgquery Commands

7.5.1 Basic cgquery Search

The first query example searches for all sequences for a known participant. The command used is listed on
the first line of the example, staring with cgquery. Notice the returned results, or Matching Objects: = 8. The results have been truncated to only show one result. All following cgquery commands will be run
with the -v option (verbose) to ensure that output is printed to the screen (stdout), otherwise no results will
be printed:

$ cgquery "participant_id=6d72de06-232a-4983-a06c-eba6d82cb3f1"

cgquery "participant_id=6d72de06-232a-4983-a06c-eba6d82cb3f1"

============================================================================
Script Version : 2.1.7
CGHub Server : https://cghub.ucsc.edu
WebServices Interface Version : 3.0
REST Resource : /cghub/metadata/analysisDetail
QueryString : participant_id=6d72de06-232a-4983-a06c-eba6d82cb3f1
Output File : None

Matching Objects : 8

---

Analysis 1

<table>
<thead>
<tr>
<th>analysis_id</th>
<th>94a37e99-757b-4912-ad68-491ce840d720</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>live</td>
</tr>
<tr>
<td>last_modified</td>
<td>2012-05-07T01:46:47Z</td>
</tr>
<tr>
<td>upload_date</td>
<td>2012-05-07T01:30:20Z</td>
</tr>
<tr>
<td>published_date</td>
<td>2012-05-07T01:46:47Z</td>
</tr>
<tr>
<td>center_name</td>
<td>UNC-LCCC</td>
</tr>
<tr>
<td>study</td>
<td>phs000178</td>
</tr>
<tr>
<td>aliquot_id</td>
<td>63b6924e-921f-4fac-9866-dd97e996d7ac</td>
</tr>
</tbody>
</table>

files

file 1

<table>
<thead>
<tr>
<th>filename</th>
<th>UNCID_1098888.63b6924e-921f-4fac-9866-dd97e996d7ac.sorted_genome_alignments.bam</th>
</tr>
</thead>
<tbody>
<tr>
<td>filesize</td>
<td>7662115253</td>
</tr>
<tr>
<td>checksum</td>
<td>75e2bd9675db6a884a51e358523132bb</td>
</tr>
<tr>
<td>legacy_sample_id</td>
<td>TCGA-55-1594-01A-01R-0946-07</td>
</tr>
<tr>
<td>disease_abbr</td>
<td>LUAD</td>
</tr>
<tr>
<td>tss_id</td>
<td>55</td>
</tr>
<tr>
<td>participant_id</td>
<td>6d72de06-232a-4983-a06c-eba6d82c3f1</td>
</tr>
<tr>
<td>sample_id</td>
<td>2885d4b3-34a6-421d-b20c-eedad721d10a</td>
</tr>
<tr>
<td>analyte_code</td>
<td>R</td>
</tr>
<tr>
<td>sample_type</td>
<td>01</td>
</tr>
<tr>
<td>library_strategy</td>
<td>RNA-Seq</td>
</tr>
<tr>
<td>platform</td>
<td>ILLUMINA</td>
</tr>
<tr>
<td>refassem_short_name</td>
<td>HG19</td>
</tr>
</tbody>
</table>

analysis_submission_uri: https://cghub.ucsc.edu/cghub/metadata/analysisSubmission/94a37e99-757b-4912-ad68-491ce840d720

analysis_full_uri : https://cghub.ucsc.edu/cghub/metadata/analysisFull/94a37e99-757b-4912-ad68-491ce840d720

analysis_data_uri : https://cghub.ucsc.edu/cghub/data/analysis/download/94a37e99-757b-4912-ad68-491ce840d720

7.5.2 Creating a metadata.xml file with cgquery

Using the analysis_id from the previous example, we can run a more in-depth search of the metadata by added the -a and the -o to the query string and create an output file:

$ cgquery "analysis_id=94a37e99-757b-4912-ad68-491ce840d720" -a -o test.xml

cgquery "analysis_id=94a37e99-757b-4912-ad68-491ce840d720" -a -o test.xml

---

Script Version : 2.1.7
CGHub Server : https://cghub.ucsc.edu
WebServices Interface Version : 3.0
REST Resource : /cghub/metadata/analysisFull

7.5. Example cgquery Commands  30
 queryString : analysis_id=94a37e99-757b-4912-ad68-491ce840d720
output_file : test.xml

matching_objects : 1

Analysis 1

    analysis_id : 94a37e99-757b-4912-ad68-491ce840d720
    state : live
    last_modified : 2012-05-07T01:46:47Z
    upload_date : 2012-05-07T01:30:20Z
    published_date : 2012-05-07T01:46:47Z
    center_name : UNC-LCCC
    study : phs000178
    aliquot_id : 63b6924e-921f-4fac-9866-dd97e996d7ac
    files
        file 1
            filename : UNCID_1098888.63b6924e-921f-4fac-9866-dd97e996d7ac.
                        sorted_genome_alignments.bam
            filesize : 7662115253
            checksum : 75e2bd9675dbea884a51e358523132bb
            legacy_sample_id : TCGA-55-1594-01A-01R-0946-07
            disease_abbr : LUAD
            tss_id : 55
            participant_id : 6d72de06-232a-4983-a06c-eba6d82chf3f1
            sample_id : 2885d4b3-34a6-421d-b20c-eedad721d10a
            analyte_code : R
            sample_type : 01
            library_strategy : RNA-Seq
            platform : ILLUMINA
            refassem_short_name : HG19
            analysis_xml : 4610 bytes of XML
            experiment_xml : 1932 bytes of XML
            run_xml : 170 by
            analysis_detail_uri : https://cghub.ucsc.edu/cghub/metadata/analysisDetail/
                              94a37e99-757b-4912-ad68-491ce840d720
            analysis_submission_uri : https://cghub.ucsc.edu/cghub/metadata/analysisSubmission/
                              94a37e99-757b-4912-ad68-491ce840d720
            analysis_data_uri : https://cghub.ucsc.edu/cghub/data/analysis/download/
                              94a37e99-757b-4912-ad68-491ce840d720

7.5.3 Modified by Month Query

The next query example searches for all sequences that have been modified in the last 5 months for example:

$ cgquery "last_modified=[NOW-5MONTH TO NOW]"


script_version : 2.1.7
CGHub_server : https://cghub.ucsc.edu

7.5. Example cgquery Commands

31
Query in progress...

***(TRUNCATED RESULTS)***

Analysis 1811

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>analysis_id</td>
<td>edff2b4a-02d4-46f2-a60c-95ca9faef5fe</td>
</tr>
<tr>
<td>state</td>
<td>live</td>
</tr>
<tr>
<td>last_modified</td>
<td>2013-01-28T17:36:01Z</td>
</tr>
<tr>
<td>upload_date</td>
<td>2013-01-28T14:52:41Z</td>
</tr>
<tr>
<td>published_date</td>
<td>2013-01-28T17:36:01Z</td>
</tr>
<tr>
<td>center_name</td>
<td>BCM</td>
</tr>
<tr>
<td>study</td>
<td>phs000178</td>
</tr>
<tr>
<td>aliquot_id</td>
<td>113b1311-d4ef-4fc8-9a8-c74fc8398756</td>
</tr>
</tbody>
</table>

**files**

<table>
<thead>
<tr>
<th>Filename</th>
<th>Size (TB)</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCGA-CM-6164-10A-01D-1650-10_hg19_Illumina.bam</td>
<td>19.82</td>
<td>fbda31d6ff50351290237763ccf360</td>
</tr>
</tbody>
</table>

Summary of Matching Objects

- downloadable_file_count: 1811
- downloadable_file_size (TB): 19.82

State count:

- live: 1811
- suppressed: 111
- submitted: 19
- uploading: 2
- redacted: 4
- augmenting_data: 8

## NOTICE: 144 matching objects are not in a downloadable state, and were therefore not displayed. To see all matching objects
regardless of state, use the --all-states option.

7.5.4 Specific Dates/Time Range Query

In addition to searching by a month or months (as in the previous example), you could use this method to search for more specific dates and time ranges:

```
```

---

**Query in progress...**

**Analysis 1809**

- **analysis_id**: edff2b4a-02d4-46f2-a60c-95ca9faef5fe
- **state**: live
- **last_modified**: 2013-01-28T17:36:01Z
- **upload_date**: 2013-01-28T14:52:41Z
- **published_date**: 2013-01-28T17:36:01Z
- **center_name**: BCM
- **study**: phs000178
- **aliquot_id**: 113b1311-d4ef-4fcb-95a8-c74fcb398756
- **files**
  - **file 1**
    - **filename**: TCGA-CM-6164-10A-01D-1650-10_hg19_Illumina.bam
    - **filesize**: 39516002578
    - **checksum**: fbda31d16fbb50351290237763cbb360
- **legacy_sample_id**: TCGA-CM-6164-10A-01D-1650-10
- **disease_abbr**: COAD
- **tss_id**: CM
- **participant_id**: 6a4031c6-ee88-48be-81a5-a522720edd02
- **sample_id**: 25c9c4c5-e578-44f3-a96c-f8ea0f5e7ea
- **analyte_code**: D
- **sample_type**: 10
- **library_strategy**: WXS
- **refassembly**: GRCh37-lite
- **analysis_submission_url**: https://cghub.ucsc.edu/cghub/metadata/analysisSubmission/edff2b4a-02d4-46f2-a60c-95ca9faef5fe
- **analysis_full_uri**: https://cghub.ucsc.edu/cghub/metadata/analysisFull/edff2b4a-02d4-46f2-a60c-95ca9faef5fe
- **analysis_data_uri**: https://cghub.ucsc.edu/cghub/data/analysis/download/edff2b4a-02d4-46f2-a60c-95ca9faef5fe

**Summary of Matching Objects**

7.5. Example cgquery Commands
7.5.5 Wildcard Queries

This example shows how to search using the wildcard character *. Suppose you have a partial filename and you know that it is a sample_type of 01:

```
$ cgquery "filename=*exome*&sample_type=01"
```

```
cgquery "filename=*exome*&sample_type=01"
```

---

Query in progress...

```
Analysis 1016
analysis_id : fd59976e-6071-40d8-9c3c-4c4822b11d6e
state : live
last_modified : 2012-11-24T21:01:40Z
upload_date : 2010-07-01T07:00:00Z
published_date : 2011-02-04T08:00:00Z
center_name : WUGSC
study : phs000178
aliquot_id : 5423db1a-5b59-4a5b-a676-00a54570b04a
files
  file 1
    filename : TCGA-13-1507-01A-01W-0549-09_IlluminaGA-DNASeq_exome.bam
    filesize : 1245665565
    checksum : 9044d36b0596e47b58c60cf55ddd2ce8
    sample_accession : SRS009222
    legacy_sample_id : TCGA-13-1507-01A-01W-0549-09
disease_abbr : OV
tss_id : 13
```
participant_id : 502e8d02-2953-4514-a2e1-6179dd94da73
sample_id : fce7a055-f6ea-4dc3-a42f-51b8ed236ca7
analyte_code : W
sample_type : 01
library_strategy : WXS
platform : ILLUMINA
refassem_short_name : NCBI36_WUGSC_variant
analysis_submission_uri: https://cghub.ucsc.edu/cghub/metadata/analysisSubmission/fd59976f-6071-40d8-9c3c-4c4822b11d6e
analysis_full_uri : https://cghub.ucsc.edu/cghub/metadata/analysisFull/fd59976f-6071-40d8-9c3c-4c4822b11d6e
analysis_data_uri : https://cghub.ucsc.edu/cghub/data/analysis/download/fd59976f-6071-40d8-9c3c-4c4822b11d6e

Summary of Matching Objects
downloadable_file_count : 1016
downloadable_file_size (TB) : 12.05
state_count
live : 1016
suppressed : 3

# NOTICE: 3 matching objects are not in a downloadable state, and were therefore not displayed. To see all matching objects regardless of state, use the --all-states option.

7.5.6 FASTQ Queries

The following example is for searching for just FASTQ formatted files which only contain the raw reads and the associated sequence qualities. This will retrieve metadata for all FASTQ files which are in study TCGA (phs000178) and are in a live state.

Note: we use the filetype fasta rather than fastq this is due to the limitations of the SRA sequence metadata schema:

cgquery -o fastqs.xml "state=live&study=phs000178&filetype=fasta"

============================================================================
Script Version : 2.1.7
CGHub Server : https://cghub.ucsc.edu
WebServices Interface Version : 3.0
REST Resource : /cghub/metadata/analysisDetail
QueryString : state=live&study=phs000178&filetype=fasta
Output File : fastqs.xml
============================================================================
Matching Objects : 618
============================================================================
Summary of Matching Objects
downloadable_file_count : 618
downloadable_file_size (TB) : 2.91
state_count
live : 618

7.5. Example cgquery Commands
7.6 CGHub REST API

The table below describes the query APIs available from CGHub.

**AnalysisDetail**

Returns minimal fields to identify the desired object as well as a URI to download the data files:

```
cghub/metadata/analysisDetail?{querystring}
```

**AnalysisFull**

Returns the full Analysis Detail attributes including the sample information and submission XML objects. The corresponding option for cgquery is `-a`:

```
cghub/metadata/analysisFull?{querystring}
```

Details of the `{querystring}` are described in section 6.1 Query Attributes.

The CGHub REST API returns query results in XML format.

The results will be saved to a file, if the `--o <filename>` option is used. This option is useful if you would like to pass a manifest.xml file to download multiple data files in one instance of gtdownload.

Example:

```
cgquery "library_strategy=WXS&sample_type=01" -o exome_test_file.xml
```

The `analysisDetail` query returns a balanced set of attributes to locate files for download. This format can be used to identify a list of sequences to pass as input to GeneTorrent to perform the transfer. It will return the following attributes to the screen:

<table>
<thead>
<tr>
<th>attribute</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>analysis_id</td>
<td>state</td>
</tr>
<tr>
<td>last_modified</td>
<td>upload_date</td>
</tr>
<tr>
<td>published_date</td>
<td>center_name</td>
</tr>
<tr>
<td>study</td>
<td>aliquot_id</td>
</tr>
<tr>
<td>files</td>
<td>filename</td>
</tr>
<tr>
<td>filesize</td>
<td>checksum</td>
</tr>
<tr>
<td>legacy_sample_id</td>
<td>disease_abbr</td>
</tr>
<tr>
<td>tss_id</td>
<td>participant_id</td>
</tr>
<tr>
<td>sample_id</td>
<td>analyte_code</td>
</tr>
<tr>
<td>sample_type</td>
<td>library_strategy</td>
</tr>
<tr>
<td>refassem_short_name</td>
<td>analysis_submission_uri</td>
</tr>
<tr>
<td>analysis_data_uri</td>
<td>analysis_full_uri</td>
</tr>
</tbody>
</table>

Example:

```
cgquery "library_strategy=WXS&sample_type=01" -o exome_test_file.xml
```
The analysisFull (invoked by adding the -a to the cgquery string) returns all of the analysisDetail attributes, except for the analysis_full_uri attribute, which is replaced by analysis_detail_uri, and will add the following three additional XML file listings:

analysis.xml, run.xml, and experiment.xml.

The analysisFull query is useful if you would like to create an output.xml file to examine the metadata associated with the data files in the query results. To create an output file that you can examine, add the -o to your query string followed by a file name of your choosing. For example:

cgquery "library_strategy=WXS&sample_type=01" -a -o exome_test_file.xml

### 7.7 Sample REST Calls

The queries will return XML formatted data, including a URL to download the associated data files for each returned analysis object.

For example, enter the following query into a web browser:

https://cghub.ucsc.edu/cghub/metadata/analysisObject?aliquot_id=c0cfafbc-6d07-4ed5-bfde-f5c3bf8437f6

or execute the command line:

$ wget --no-check-certificate -O output.xml https://cghub.ucsc.edu/cghub/metadata/analysisObject?aliquot_id=c0cfafbc-6d07-4ed5-bfde-f5c3bf8437f6

The returned XML will look something like the result on the next page:

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<ResultSet date="2012-01-09 09:19:22">
  <Query>aliquot_id:c0cfafbc-6d07-4ed5-bfde-f5c3bf8437f6</Query>
  <Hits>1</Hits>
  <Result id="1">
    <analysis_id>8d0b3af6-0f78-4282-bde5-c5238fa3d4c2</analysis_id>
    <state>live</state>
    <last_modified>2011-08-10T07:00:00Z</last_modified>
    <upload_date></upload_date>
    <center_name>BCCAGSC</center_name>
    <study>phs000178</study>
    <aliquot_id>c0cfafbc-6d07-4ed5-bfde-f5c3bf8437f6</aliquot_id>
    <files>
      <file>
        <filename>TCGA-CZ-5466-11A-01R-1502-13_mirna.bam</filename>
        <filesize>132976422</filesize>
        <checksum type="MD5">6ede057c98171b986a36e77a9bf7efc0</checksum>
      </file>
    </files>
    <analysis_attribute_uri>https://cghub.ucsc.edu/cghub/metadata/analysisAttributes/8d0b3af6-0f78-4282-bde5-c5238fa3d4c2</analysis_attribute_uri>
    <analysis_data_uri>https://cghub.ucsc.edu/cghub/data/analysis/download/8d0b3af6-0f78-4282-bde5-c5238fa3d4c2</analysis_data_uri>
  </Result>
</ResultSet>
```

7.7. Sample REST Calls 37
The results return the same attribute fields that a cgquery will return (echo) to your screen, except that they are in XML format when viewed from a URL or output file (-o). You may add the "-a" and "-o" to create a full metadata.xml file as well.

7.7.1 Paging

To limit the number of results displayed per “page” or to contain large query manifest.xml files use the “start” and “rows” options.

https://cghub.ucsc.edu/cghub/metadata/analysisDetail?state=live&start=0&rows=100 or:

cgquery -o t1.xml "state=live&start=0&rows=100"

“start” starts at the index of 0.

“rows” will be the number of results to display.

In order to page the first one-hundred items use: "...start=0&rows=100"

To page the next hundred items use: "...start=100&rows=100"

Keep in mind that the index starts at “0”. To build a manifest.xml file with an odd number, say 15 items in 4 manifest.xml files, to obtain the first 60 items use this example:

"...start=0&rows=15"

"...start=15&rows=15"

"...start=30&rows=15"

"...start=45&rows=15"

7.7.2 JSON (JavaScript Object Notation) Instead of XML Files

Use a normal URI call to the CGHub database and add the -H option:


In this example we searched for FASTQs because all “*.tar” filenames are FASTQ files.
The Annai GeneTorrent download client has been provided as the means to download sequence data files from the CGHub system securely, efficiently, and reliably. The Annai GeneTorrent software an open-source application available at no cost to users.

8.1 Downloading Files using gtdownload (GeneTorrent Client)

The gtdownload client runs as a command line tool on all systems.

The download mode, minimum command line arguments are:

1. gtdownload
2. -d [UUID | .xml]
3. -c <credentials file>, Example:/home/path/to/cghub.key or https://www.cghub.ucsc.edu/software/downloads/cghub_public.key

Note: If no path is specified [-p /path/to/dir/to/save/data], the current working directory is the default storage location.

For example, using a manifest.xml to download publicly accessible data:

    gtdownload -c https://www.cghub.ucsc.edu/software/downloads/cghub_public.pem -d manifest.xml -vv

Similarly, to download secure data:

gtdownload -vv -c cghub.key -d manifest.xml

Download publicly accessible data by explicitly specifying the UUID of a submission:

    gtdownload -d ebd53ae-6386-4bc4-90b1-4f249ff9fcdf -c cghub_public.key -vv

Similarly, to download secure data use your secure key file:

    gtdownload -d 5d9f808-2d44-4db3-9336-ede7680d1eaf -c /home/path/to/cghub.key -vv
Some useful arguments when using gtdownload are listed below. A full listing of arguments can be found by “man gtdownload” on a command line for Linux and Mac users, and Windows users can find this list in the gtdownload_manual.txt located in the GeneTorrent folder.

-d content-specifier
The content-specifier should be one of the following:

1. A UUID denoting an analysis object at the GeneTorrent Executive. In this case, gtdownload will construct a URI based on the default server, currently https://cghub.ucsc.edu.

Note: Multiple -d options can be specified on the same command line, in which case multiple analysis objects will be downloaded.

2. An XML file, which will be parsed to obtain a list of URIs.

-c credential-file
A credential file is required for download mode, unless you are directly passing a .gto file. Pass the full or relative path to the file containing the access credentials (security key) previously received from the User Authentication, (see section 4 for details).

-p path
Path to save data files in the gto file(s). The analysis UUID will always be added to path, so data files will be found at path/UUID. The current directory will be used by default.

-C confdir
This argument has been deprecated. Please use -R instead.

-R resourcedir
Specify an absolute path to the GeneTorrent resource directory. This directory should contain the dhparam.pem file.

This option is only required if GeneTorrent was installed using a non-standard directory layout. The default path to the resource directory is derived at build time from the value of the --prefix configure option. The GTRESOURCE_DIR environment variable can be set to override that default.

-l destination[:verbosity]
Specifies the destination of a log file and a verbosity level. Possible values for the destination include none, stdout, stderr, syslog, or a filename; the log output will be directed to the destination specified. Possible values for verbosity include (in order of increasing volume of messages) standard, verbose, or full; standard is the default. The default behavior is no logging if this parameter is not specified.

-v, -vv
Controls the level of on-screen progress reporting to stdout. By default GeneTorrent offers no status information unless an error occurs. You can receive summary progress information by specifying a single -v, or detailed progress information by specifying –vv e.g., two -v’s.
-t  
Specifies that the on-screen progress messages (controlled by the -v flag, above) should include a timestamp. This is primarily useful when these messages are being redirected to a file for later review.

-k minutes  
Specifies a duration of inactivity in minutes after which GeneTorrent, in the download or upload application, will abort file transfers and exit. The activity measured to make this determination is actual payload data transfer over the GeneTorrent protocol. The default behavior is to never timeout, which is the recommended setting. When using this setting, it is recommended that it be set for 40 or more minutes.

-max-children n  
The maximum number of parallel children (n) that should be spawned to perform the download. By default, 8 children are used, but this number may be adjusted. In general, you should have equivalent or fewer children than you have CPU cores on your machine. See the GeneTorrent “man page” for the most up to date description.

-rate-limit n  
This limits the network transfer rate of GeneTorrent where the maximum rate is an integer (n) specified in Millions of Bytes (MB).

-null-storage  
A virtual storage device intended for a receive-only peer, which throws away all data received (rather than writing to disk). This mode disables hash checking of the downloaded pieces so there is no handling of corrupt pieces, every piece downloaded is assumed correct. This is very useful for testing network performance.

8.2 Query and Download

Step 1: Use cgquery to find the desired files:

$ cgquery "center_name=(BI OR BCM)&disease_abbr=LUAD"

============================================================================
Script Version : 2.1.7
CGHub Server : https://cghub.ucsc.edu
WebServices Interface Version : 3.0
REST Resource : /cghub/metadata/analysisDetail
QueryString : center_name=(BI OR BCM)&disease_abbr=LUAD
Output File : None
----------------------------------------------------------------------------
Query in progress...

1013 results truncated and this example is only listing the last result echoed to the screen.

Analysis 1013
  analysis_id : 5d97f808-2d44-4db3-9336-ede7680d1eaf
  state : live
  last_modified : 2012-12-26T00:22:08Z
  upload_date : 2012-12-25T22:08:36Z
Step 2: Invoke gtdownload on the command line with the desired file UUIDs or manifest.xml file (-d), and user credentials (-c):

```bash
$ gtdownload -v -c /path/to/cghub.key -d 5d97f808-2d44-4db3-9336-edf7650d1eaf
```

or

```bash
$ gtdownload -v -c /path/to/cghub.key –d manifest.xml
```

The GeneTorrent client connects to the CGHub system, bi-directionally authenticates using the credential, verifies that the user is authorized to access the file, and starts the SSL-secured file transfer.

Note:

gtdownload, by default runs silently to completion. If invoked with the -v (verbose) option, the user will see something similar to the example below:
$ gtdownload -c cghub.key -d 5d97f808-2d44-4db3-9336-ede7680d1eaf -v

Welcome to GeneTorrent version 3.8.0, download mode.
Ready to download
Communicating with GT Executive ...
Headers received from the client: 'HTTP/1.1 100 Continue
HTTP/1.1 200 OK
Date: Mon, 28 Jan 2013 23:39:28 GMT
Server: Apache/2.2.15 (CentOS)
Strict-Transport-Security: max-age=31536000
X-Powered-By: PHP/5.3.2
Content-Description: File Transfer
Content-Disposition: attachment; filename=a9d4e154-69f5-4558-a345-cd7c35a383a2.gto
Content-Transfer-Encoding: binary
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Content-Length: 77576
Connection: close
Content-Type: application/octet-stream

* About to connect() to cghub.ucsc.edu port 443 (#0)
* Trying 192.35.223.5... * connected
* successfully set certificate verify locations:
  * CAfile: none
  * CApath: /etc/ssl/certs
* SSL connection using DHE-RSA-AES256-SHA
* About to connect() to cghub.ucsc.edu port 443 (#0)
* Trying 192.35.223.5... * connected
* successfully set certificate verify locations:
  * CAfile: none
  * CApath: /etc/ssl/certs
* SSL connection using DHE-RSA-AES256-SHA
* About to connect() to cghub.ucsc.edu port 443 (#0)
* Trying 192.35.223.5... * connected
* successfully set certificate verify locations:
  * CAfile: none
  * CApath: /etc/ssl/certs
* SSL connection using DHE-RSA-AES256-SHA
* About to connect() to cghub.ucsc.edu port 443 (#0)
* Trying 192.35.223.5... * connected
* successfully set certificate verify locations:
  * CAfile: none
  * CApath: /etc/ssl/certs
* SSL connection using DHE-RSA-AES256-SHA
* About to connect() to cghub.ucsc.edu port 443 (#0)
* Trying 192.35.223.5... * connected
* successfully set certificate verify locations:
  * CAfile: none
  * CApath: /etc/ssl/certs
* SSL connection using DHE-RSA-AES256-SHA
* About to connect() to cghub.ucsc.edu port 443 (#0)
* Trying 192.35.223.5... * connected
* successfully set certificate verify locations:
  * CAfile: none
  * CApath: /etc/ssl/certs
* SSL connection using DHE-RSA-AES256-SHA

> POST /cghub/data/gtsession HTTP/1.1
Host: cghub.ucsc.edu
Accept: */*
Content-Length: 1237
Expect: 100-continue
Content-Type: multipart/form-data; boundary=-----------------------d417fc64a39e

< HTTP/1.1 100 Continue
d417fc64a39e
< HTTP/1.1 200 OK
< Date: Mon, 28 Jan 2013 23:39:29 GMT
< Server: Apache/2.2.15 (CentOS)
< Strict-Transport-Security: max-age=31536000
< X-Powered-By: PHP/5.3.2
< Content-Length: 1375
< Connection: close
< Content-Type: text/html; charset=UTF-8
< Closing connection #0

Headers received from the client: 'HTTP/1.1 100 Continue
HTTP/1.1 200 OK
Step 3: Verify that the download was successful.

When GeneTorrent finishes transferring the file, the user will find the data file in the current working directory, or in the location specified by the “-p” (path) option.

The file length and the MD5 checksum of the file can be verified using standard tools provided by the operating system.

### 8.3 Query and Download using cgquery Interactive Mode

The cgquery –i option will prompt the user to download one or more of the returned files using gtdownload (GeneTorrent). The interactive command requires the user credentials. In the case of a custom install location, the command needs some additional information regarding the location of gtdownload (GeneTorrent):

```bash
cgquery "filename=*AR-A0TX*" -i -c cghub.key -v -g /path/to/cghub/bin/gtdownload -f /path/to/cghub/share/GeneTorrent
```

The command when gtdownload is in the default location (/usr/bin) is shown below:

```bash
$ cgquery "filename=*AR-A0TX*" -i -c cghub.key -v
cgquery "filename=*AR-A0TX*" -i -c cghub.key -v
```

---

<table>
<thead>
<tr>
<th>Script Version</th>
<th>2.1.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGHub Server</td>
<td><a href="https://cghub.ucsc.edu">https://cghub.ucsc.edu</a></td>
</tr>
<tr>
<td>WebServices Interface Version</td>
<td>3.0</td>
</tr>
<tr>
<td>REST Resource</td>
<td>/cghub/metadata/analysisDetail</td>
</tr>
<tr>
<td>QueryString</td>
<td>filename=<em>AR-A0TX</em></td>
</tr>
<tr>
<td>Output File</td>
<td>None</td>
</tr>
</tbody>
</table>

---

Result 1
Analysis 1
```
analysis_id : cacbac47-8c27-45e6-ae3d-96653a6a5768
```
<table>
<thead>
<tr>
<th>file 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>filename : TCGA-AR-A0TX-01A-11R-A085-13_mirna.bam</td>
</tr>
<tr>
<td>filesize : 111802454</td>
</tr>
<tr>
<td>checksum : 6344810f578e1bfff01ad4402f1d2b40</td>
</tr>
<tr>
<td>sample_accession : SRS133274</td>
</tr>
<tr>
<td>legacy_sample_id : TCGA-AR-A0TX-01A-11R-A085-13</td>
</tr>
<tr>
<td>disease_abbr : BRCA</td>
</tr>
<tr>
<td>tss_id : AR</td>
</tr>
<tr>
<td>participant_id : c364e81c-eble-4870-9c661f5f2e3d</td>
</tr>
<tr>
<td>sample_id : d5adaf35-04b1-4198-b308-f4720b3d6417</td>
</tr>
<tr>
<td>analyte_code : R</td>
</tr>
<tr>
<td>sample_type : 01</td>
</tr>
<tr>
<td>library_strategy : miRNA-Seq</td>
</tr>
<tr>
<td>platform : ILLUMINA</td>
</tr>
<tr>
<td>refassem_short_name : GRCh37-lite</td>
</tr>
</tbody>
</table>

analysis_submission_uri : https://cghub.ucsc.edu/cghub/metadata/analysisSubmission/cacb4c7-8c27-45e6-ae3d-96653a6a5768

analysis_full_uri : https://cghub.ucsc.edu/cghub/metadata/analysisFull/cacb4c7-8c27-45e6-ae3d-96653a6a5768

analysis_data_uri : https://cghub.ucsc.edu/cghub/data/analysis/download/cacb4c7-8c27-45e6-ae3d-96653a6a5768

Results 2 and 3 truncated ...

Summary of Matching Objects

| downloadable_file_count : 3 |
| downloadable_file_size (GB) : 16.56 |
| state_count |
| live : 3 |

All matching objects are in a downloadable state.

Enter the index of a URI to Denise Hamilton download (0 for all), a range of indeces separated by dash, or 'q' to quit

[ 0 ] : All URIs
[ 1 ] : https://cghub.ucsc.edu/cghub/data/analysis/download/cacb4c7-8c27-45e6-ae3d-96653a6a5768
[ 2 ] : https://cghub.ucsc.edu/cghub/data/analysis/download/03a87c60-2688-4be6-8f11-ce109e96955b
[ 3 ] : https://cghub.ucsc.edu/cghub/data/analysis/download/e3bfe3bb-d92e-4c5a-ab7a-9ed37503201a

Index (0=all, q=quit) > 1-2
At this point, you can enter 1, 2, or 3 to download individually, or 0 for all results listed. You can also select a range of analysis_id’s to download.

In this example: 1-2 was the selected range, which will begin to download UUIDs 1 and 2. Once UUIDs 1 and 2 have been downloaded you will be returned to this screen were you may select and download result 3 (or additional results if present), or quit (q).
INDEX

Symbols

–max-children n
  gtdownload command line option, 41
–null-storage
  gtdownload command line option, 41
–rate-limit n
  gtdownload command line option, 41
–version
  cgquery command line option, 28
-A, –all-states
  cgquery command line option, 29
-C confdir
  gtdownload command line option, 40
-I, –ident
  cgquery command line option, 28
-R resourcedir
  gtdownload command line option, 40
-S SORTBY, –sort=SORTBY
  cgquery command line option, 28
-a, –attributes
  cgquery command line option, 29
-b SUBMISSION, –submission=SUBMISSION
  cgquery command line option, 29
-c CREDENTIAL, –credential=CREDENTIAL
  cgquery command line option, 29
-c credential-file
  gtdownload command line option, 40
-d content-specifier
  gtdownload command line option, 40
-f CONFDIR, –conf-dir=CONFDIR
  cgquery command line option, 29
-g GTBIN, –gt-binary=GTBIN
  cgquery command line option, 28
-h, –help
  cgquery command line option, 28
-i, –interactive
  cgquery command line option, 28
  gtdownload command line option, 41
-l destination[:verbosity]
  gtdownload command line option, 40
-n, –count-only (default)
  cgquery command line option, 29
-o OUTPUTXML, –outputxml=OUTPUTXML
  cgquery command line option, 28
-p path
  gtdownload command line option, 40
-s SERVER, –server=SERVER
  cgquery command line option, 28
-t
  gtdownload command line option, 40
-v or –vv, –verbose
  cgquery command line option, 29
-v, -vv
  gtdownload command line option, 40

C

cgquery command line option
  –version, 28
-A, –all-states, 29
-I, –ident, 28
-S SORTBY, –sort=SORTBY, 28
-a, –attributes, 29
-b SUBMISSION, –submission=SUBMISSION
  submission=SUBMISSION, 29
-c CREDENTIAL, –credential=CREDENTIAL
  credential=CREDENTIAL, 29
-f CONFDIR, –conf-dir=CONFDIR
  gtdownload command line option, 29
-g GTBIN, –gt-binary=GTBIN
  gtdownload command line option, 28
-h, –help, 28
-i, –interactive, 29
-n, –count-only (default), 29
-o OUTPUTXML, outputxml=OUTPUTXML, 28
-s SERVER, server=SERVER, 28
-v or –vv, –verbose, 29

G
gtdownload command line option
- max-children n, 41
- null-storage, 41
- rate-limit n, 41
- C confdir, 40
- R resourcedir, 40
- c credential-file, 40
- d content-specifier, 40
- k minutes, 41
- l destination[:verbosity], 40
- p path, 40
- t, 40
- v, -vv, 40