GDC Data Transfer Tool User's Guide

NCI Genomic Data Commons (GDC)

Contents

Chapter 1

Getting Started

Getting Started

The GDC Data Transfer Tool: An Overview

Raw sequence data, stored as BAM files, make up the bulk of data stored at the NCI Genomic Data Commons (GDC). The size of a single file can vary greatly. Most BAM files stored in the GDC are in the 50 MB - 40 GB size range, with some of the whole genome BAM files reaching sizes of 200-300 GB.

The GDC Data Transfer Tool, a command-line driven application, provides an optimized method of transferring data to and from the GDC and enables resumption of interrupted transfers.

Downloading the GDC Data Transfer Tool

System Recommendations

The system recommendations for using the GDC Data Transfer Tool are as follows:

- OS: Linux (Ubuntu 14.x or later), OS X (10.9 Mavericks or later), or Windows (7 or later)
- CPU: At least eight 64-bit cores, Intel or AMD
- RAM: At least 8 GiB
- Storage: Enterprise-class storage system capable of at least 1 Gb/s (gigabit per second) write throughput and sufficient free space for BAM files.

Binary Distributions

Binary distributions are available on the GDC Transfer Tool page. To install the GDC Data Transfer Tool, download the respective binary distribution and unzip the distribution's archive to a location on the target system.

Release Notes

Release Notes are available on the GDC Data Transfer Tool Release Notes Page.

Chapter 2

Accessing Built-in Help

Help Menus

The GDC Data Transfer Tool comes with built-in help menus. These menus are displayed when the GDC Data Transfer Tool is run with flags -h or –help for any of the main arguments to the tool. Running the GDC Data Transfer Tool without argument or flag will present a list of available command options.

```
gdc-client --help
```

```
usage: gdc-client [-h] [--version] {download,upload,interactive} ...

The Genomic Data Commons Command Line Client

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit

commands:
  {download,upload,interactive}
                        for more information, specify -h after a command
    download            download data from the GDC
    upload              upload data to the GDC
    interactive         run in interactive mode
```

The available menus are provided below.

Root menu

The GDC Data Transfer Tool displays the following output when executed without any arguments.

```
gdc-client
```

```
usage: gdc-client [-h] [--version] {download,upload,interactive} ...
gdc-client: error: too few arguments
```

Download help menu

The GDC Data Transfer Tool displays the following help menu for its download functionality.

```
gdc-client download --help
```

```
usage: gdc-client download [-h] [--debug] [-v] [--log-file LOG_FILE]
                           [-T TOKEN | -t TOKEN] [-H HOST] [-P PORT] [-d DIR]
                           [-s server] [--no-segment-md5sums] [-n N_PROCESSES]
                           [--http-chunk-size HTTP_CHUNK_SIZE]
                           [--save-interval SAVE_INTERVAL]
                           [--no-related-files] [--no-annotations] [-u]
                           [-m MANIFEST]
                           [file_id [file_id ...]]

positional arguments:
  file_id               GDC files to download

optional arguments:
  -h, --help            show this help message and exit
  --debug               enable debug logging
  -v, --verbose         enable verbose logging
  --log-file LOG_FILE   log file [stderr]
  -t TOKEN, --token-file TOKEN
                        GDC API auth token file
  -H HOST, --host HOST  GDC API host [gdc-api.nci.nih.gov]
  -P PORT, --port PORT  GDC API port [443]
  -d DIR, --dir DIR     Directory to download files to. Defaults to current
                        dir
  -s server, --server server
                        The TCP server address server[:port]
  --no-segment-md5sums  Calculate inbound segment md5sums and/or verify
                        md5sums on restart
  -n N_PROCESSES, --n-processes N_PROCESSES
                        Number of client connections.
  --http-chunk-size HTTP_CHUNK_SIZE
                        Size in bytes of standard HTTP block size.
  --save-interval SAVE_INTERVAL
                        The number of chunks after which to flush state file.
                        A lower save interval will result in more frequent
                        printout but lower performance.
  --no-related-files    Do not download related files.
  --no-annotations      Do not download annotations.
  -u, --udt             Use the UDT protocol. Better for WAN connections
  -m MANIFEST, --manifest MANIFEST
                        GDC download manifest file
```

Upload help menu

The GDC Data Transfer Tool displays the following help menu for its upload functionality.

```
gdc-client upload --help
```

```
usage: gdc-client upload [-h] [--debug] [-v] [--log-file LOG_FILE]
                         [-T TOKEN | -t TOKEN] [-H HOST] [-P PORT]
                         [--project-id PROJECT_ID] [--identifier IDENTIFIER]
                         [--path path] [--upload-id UPLOAD_ID] [--insecure]
                         [--server SERVER] [--part-size PART_SIZE]
                         [-n N_PROCESSES] [--disable-multipart] [--abort]
                         [--resume] [--delete] [--manifest MANIFEST]

optional arguments:
  -h, --help            show this help message and exit
```

```
  --debug               enable debug logging
  -v, --verbose         enable verbose logging
  --log-file LOG_FILE   log file [stderr]
  -t TOKEN, --token-file TOKEN
                        GDC API auth token file
  -H HOST, --host HOST  GDC API host [gdc-api.nci.nih.gov]
  -P PORT, --port PORT  GDC API port [443]
  --project-id PROJECT_ID, -p PROJECT_ID
                        The project ID that owns the file
  --identifier IDENTIFIER, -i IDENTIFIER
                        The file id
  --path path, -f path  directory path to find file
  --upload-id UPLOAD_ID, -u UPLOAD_ID
                        Multipart upload id
  --insecure, -k        Allow connections to server without certs
  --server SERVER, -s SERVER
                        GDC API server address
  --part-size PART_SIZE, -ps PART_SIZE
                        Part size for multipart upload
  -n N_PROCESSES, --n-processes N_PROCESSES
                        Number of client connections
  --disable-multipart   Disable multipart upload
  --abort               Abort previous multipart upload
  --resume, -r          Resume previous multipart upload
  --delete              Delete an uploaded file
  --manifest MANIFEST, -m MANIFEST
                        Manifest which describes files to be uploaded
```

Chapter 3

Preparing for Data Download and Upload

Preparing for Data Downloads and Uploads

Overview

The GDC Data Transfer Tool is intended to be used in conjunction with the GDC Data Portal and the GDC Data Submission Portal to transfer data to or from the GDC. First, the GDC Data Portal's interface is used to generate a manifest file or obtain UUID(s) and (for Controlled-Access Data) an authentication token. The GDC Data Transfer Tool is then used to transfer the data files listed in the manifest file or identified by UUID(s).

Downloads

Obtaining a Manifest File for Data Download

The GDC Data Transfer Tool supports downloading multiple files listed in a GDC manifest file. Manifest files can be generated and downloaded directly from the GDC Data Portal:

First, select the data files of interest. Click the *Cart* button in the row corresponding to the file desired. The button will turn green to indicate that the file has been selected.

Once all files of interest have been selected, click on the *Cart* button in the upper right-hand corner. This will bring up the cart page, which provides an overview of all currently selected files. This list of files can be downloaded as a manifest file by clicking on the green *Download* button and selecting *Manifest* from the drop down.

Obtaining UUIDs for Data Download

A manifest file is not required to download files from GDC. The GDC Data Transfer Tool will accept file UUID(s) instead of a manifest file for downloading individual data files. To obtain a data file's UUID from the GDC Data Portal, click the file name to find its detail page including its GDC UUID.

Obtaining an Authentication Token for Data Downloads

The GDC Data Transfer Tool requires an authentication token to download from GDC data portal to download Controlled-Access Data. Tokens can be generated and downloaded directly from the GDC Data Portal.

To generate a token, first log in to the GDC Data Portal by clicking the *Login* button in the top right corner of the page. This will redirect to the eRA Commons login page. After successful authentication, the GDC Data Portal will display the username in place of the *Login* button. Here, the user Ian Miller is logged in to the GDC Data Portal, indicated by the username IANMILLER.

Clicking the username will open a drop-down menu. Select *Download Token* from the menu to generate an authentication token.

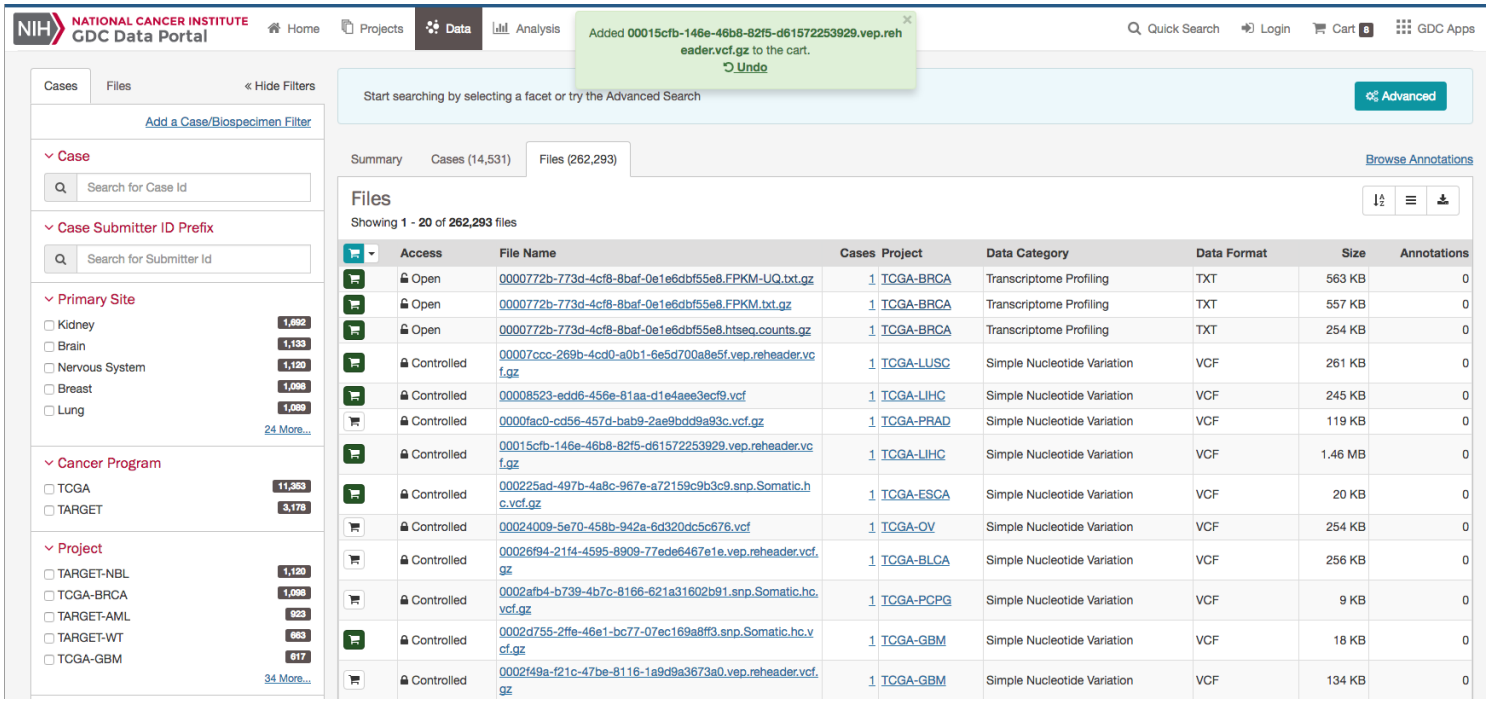


Figure 3.1: GDC Data Portal: Selecting Files of Interest

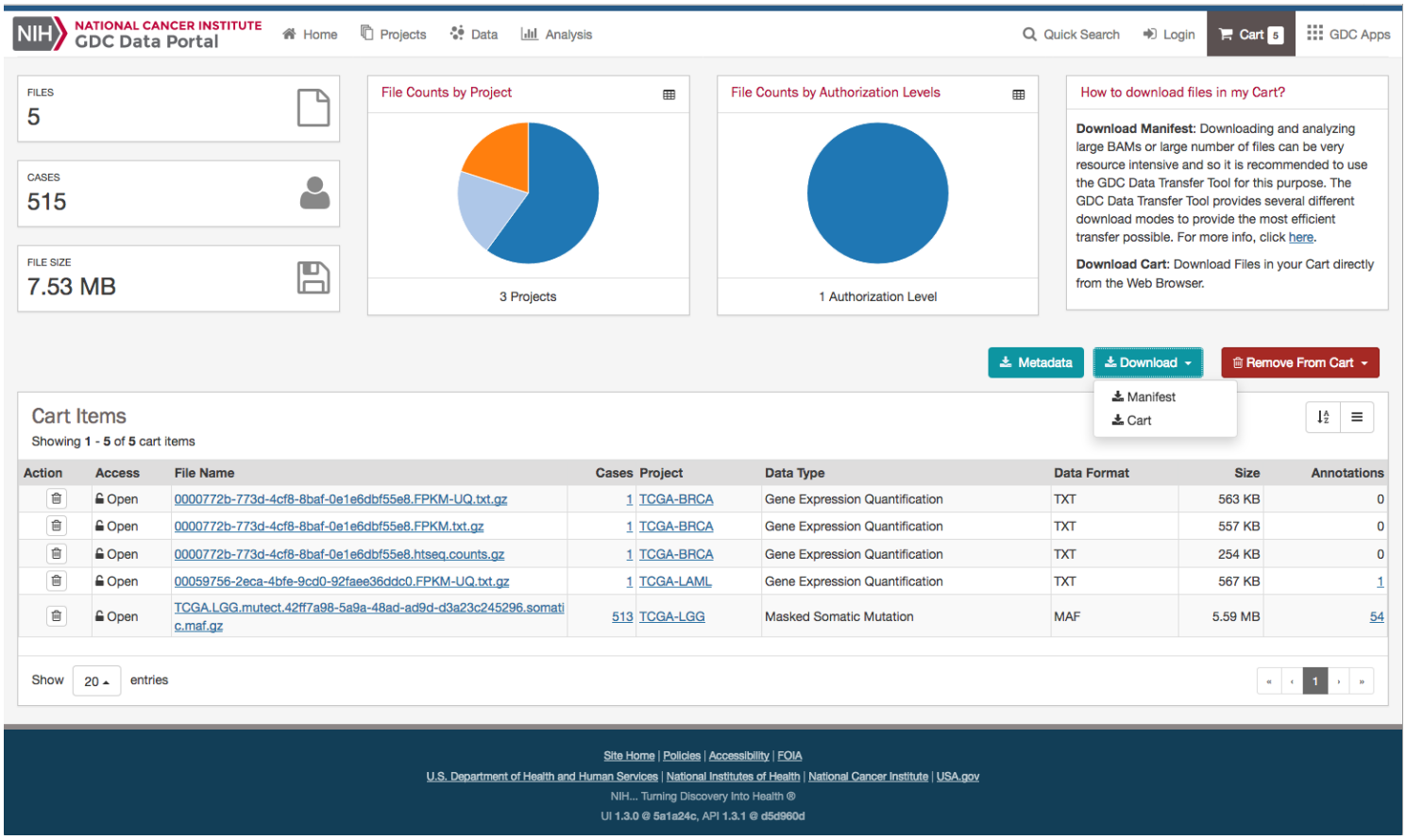


Figure 3.2: GDC Data Portal: Cart Page

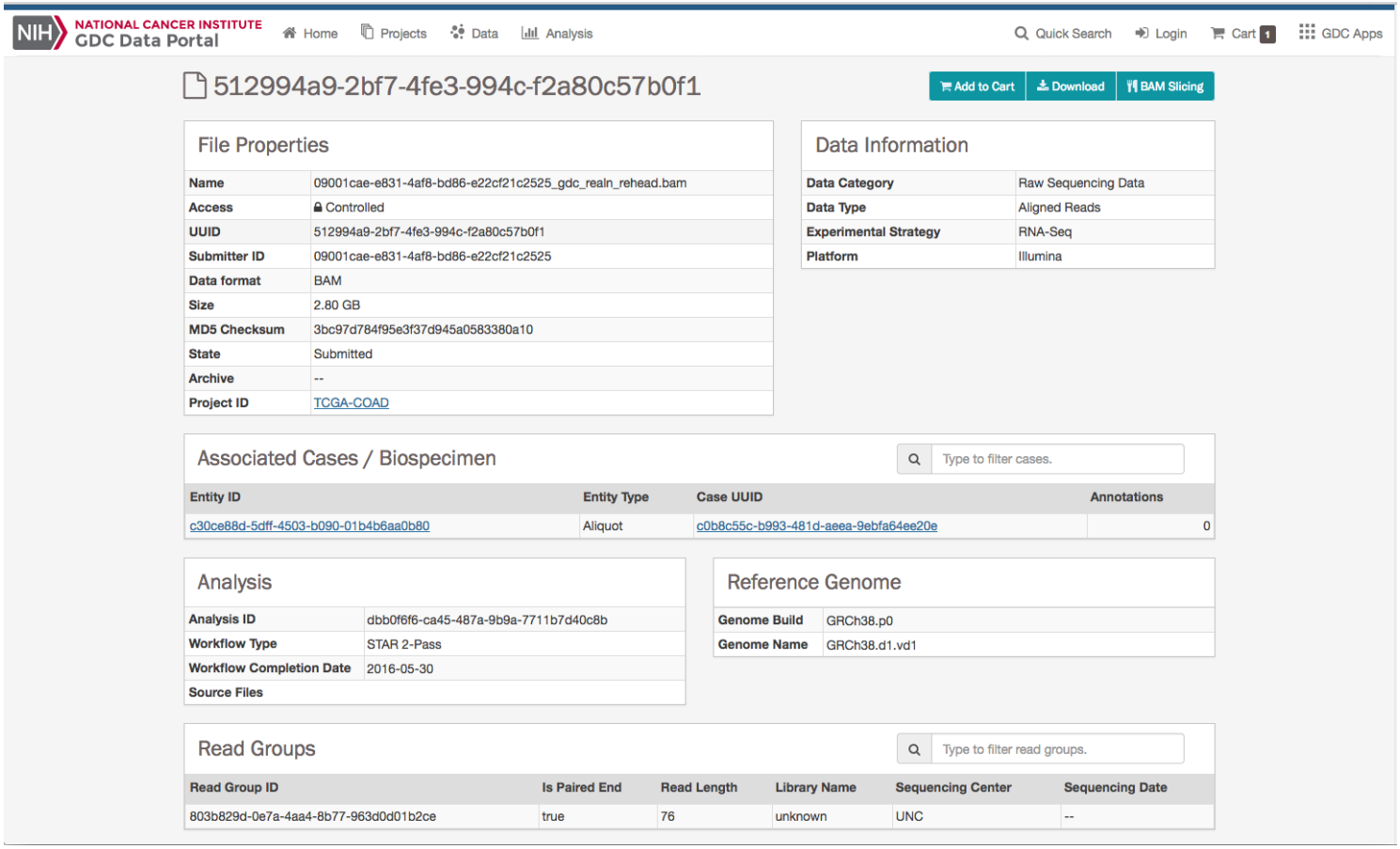
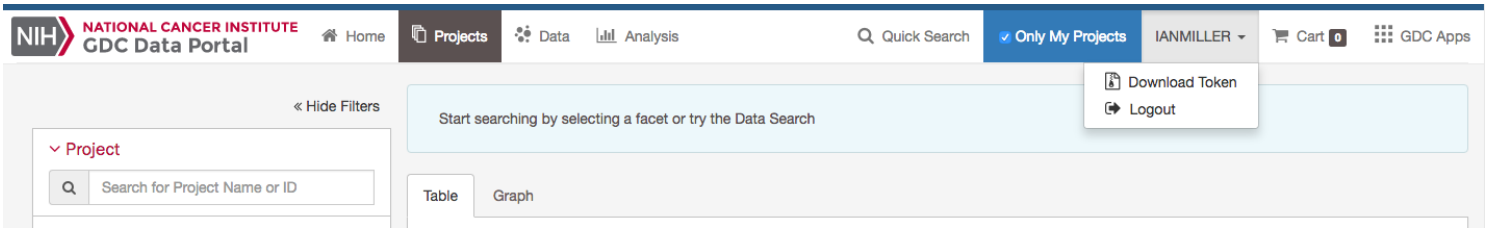


Figure 3.3: GDC Data Portal: Detailed File Page

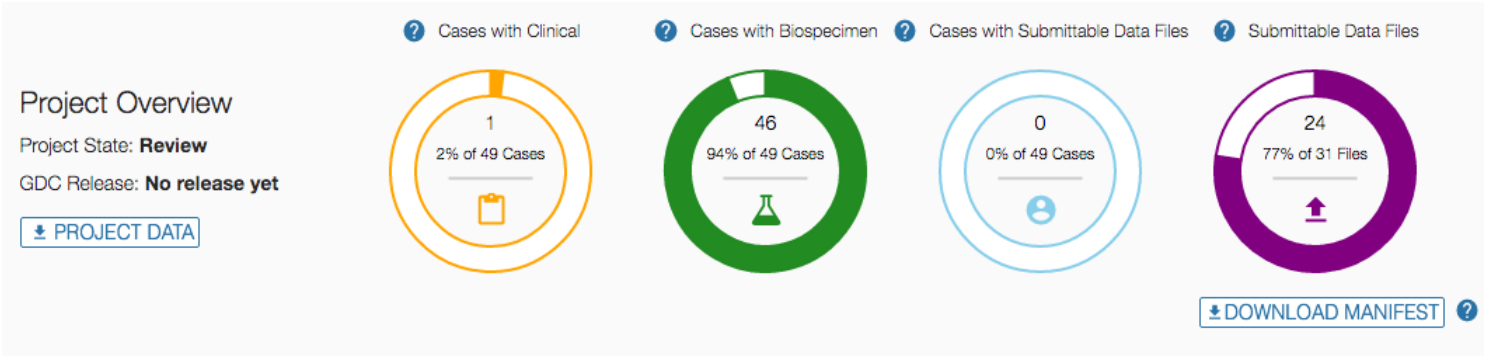


NOTE: The authentication token should be kept in a secure location, as it allows access to all data accessible by the associated user.

Uploads

Obtaining a Manifest File for Data Uploads

Multiple data file uploads are supported by the GDC Data Transfer Tool via a manifest file. Manifest files can be generated and downloaded directly from the GDC Submission Portal. A project's manifest file can be downloaded from the projects's dashboard.



NOTE: To download a project's manifest file click on the *Download Manifest* button located on the home page of the project, just below the four status charts. A manifest will be generated for the entire project or if previous files have already been upload only the files that remain to be uploaded.

A manifest for individual files can also be downloaded from the transaction tab and browse tab pages of the submission portal's project. More information on the process can be found under the Submission Portal's documentation section entitled Step 4: GDC Data Transfer Tool.

Obtaining UUIDs for Data Uploads

A UUID can be used for data submission with the Data Transfer Tool. The UUID for submittable data uploads can be obtained from the Submission Portal or from the API GraphQL endpoint. In the Submission Portal the UUID for a data file can be found in the Manifest YAML file located in the *id:* row located under the file size entry.

A second location to obtain a UUID in the Submission Portal is on the Browse Tab page. Under the Submittable Data Files section a UUID can be found by opening up the file's detail page. By clicking on the Submitter ID of the upload file a new window will display a Summary of the file's details, which contains the UUID.

GraphQL A UUID can be obtained from the API GraphQL endpoint. An overview of what GraphQL and its uses is located on the API documentation page section Querying Submitted Data Using GraphQL

The following example will query the endpoint to produce a UUID along with submitter_id, file_name, and project_id.

```
{
  submitted_unaligned_reads (project_id: "GDC-INTERNAL", submitter_id:
      "Blood-00001-aliquot_lane1_barcode23.fastq") {
   id
   submitter_id
   file_name
```

```
files:
- data_category: Raw Sequencing Data
  data_format: FASTQ
  data_type: Unaligned Reads
  experimental_strategy: WGS
  file_name: GDC-INTERNAL-000084-S1-Q1-RG1.fastq.zip
  file_size: 430112000
  id: c414a205-376e-4993-af48-2a4689eb433e
  local_file_path: GDC-INTERNAL-000084-S1-Q1-RG1.fastq.zip
  md5sum: e0bb0367ffbc287dcf10ed4212a740a2
  project_id: GDC-INTERNAL
  read_groups:
  - id: 4231ef42-4f24-48f1-88da-aa98b492e57e
    submitter_id: GDC-INTERNAL-000084-S1-Q1-RG1
  state_comment: null
  submitter_id: GDC-INTERNAL-000084-S1-Q1-RG1.fastq.zip
  type: submitted_unaligned_reads
```

Figure 3.4: Submission Manifest yaml file

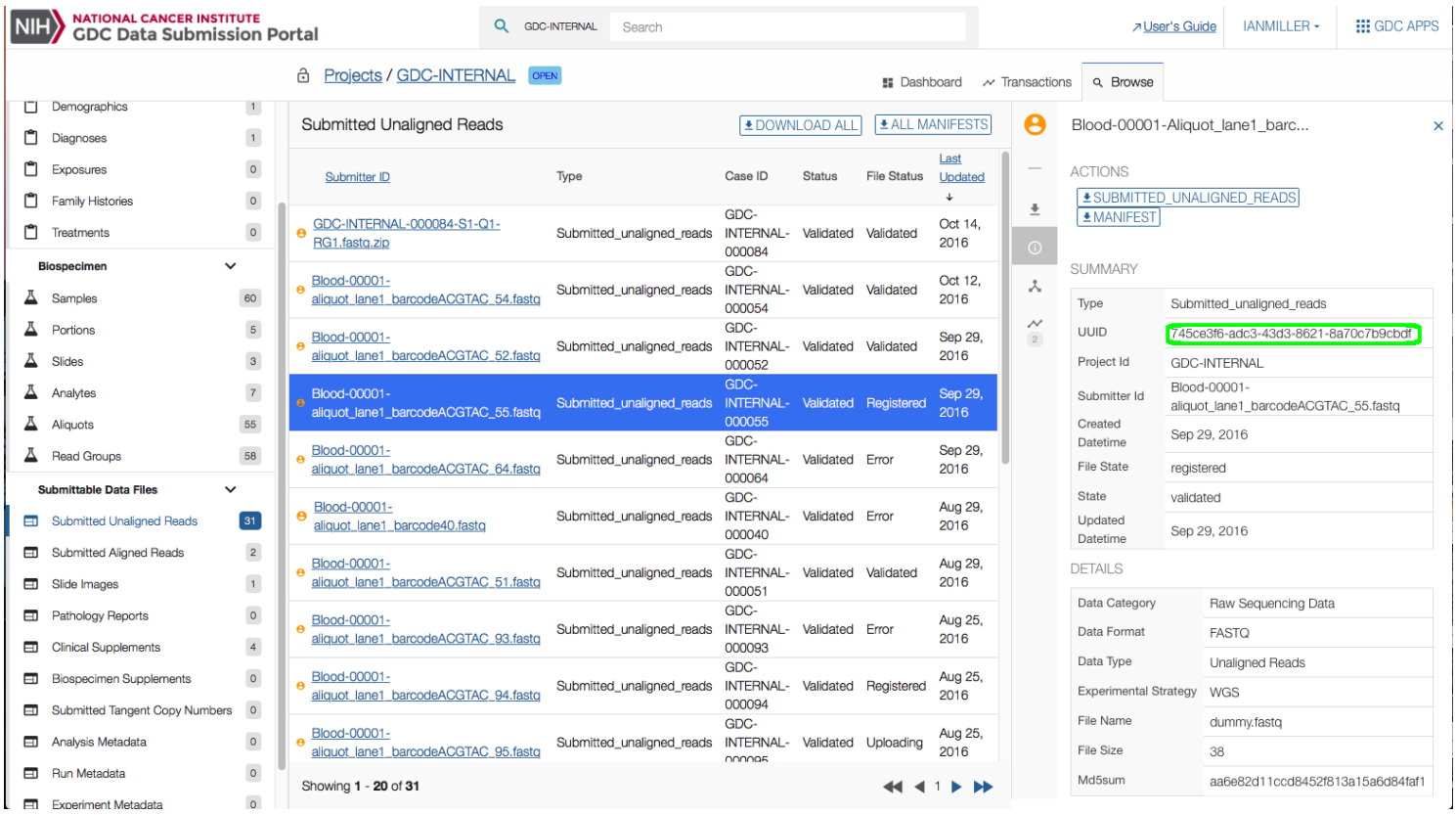


Figure 3.5: Submission Portal Browse Page Details

```
   project_id
}
}
```

```
{\n  submitted_unaligned_reads (project_id: \"GDC-INTERNAL\", submitter_id:
    \"Blood-00001-aliquot_lane1_barcode23.fastq\") {\n   id\n   submitter_id\n   file_name\n
    project_id\n}\n}\n
```

```
{
        "query": "{\n \n  submitted_unaligned_reads (project_id: \"GDC-INTERNAL\", submitter_id:
            \"Blood-00001-aliquot_lane1_barcode23.fastq\") {\n    id\n    submitter_id\n    file_name\n
               project_id\n}\n}",
        "variables": null
}
```

```
export
    token=ALPHANUMERICTOKEN-01234567890+AlPhAnUmErIcToKeN=0123456789-ALPHANUMERICTOKEN-01234567890+AlPhAnUmErIcT
$ curl --request POST --header "X-Auth-Token: $token" 'https://api.gdc.cancer.gov/v0/submission/graphql'
    -d@data.json
```

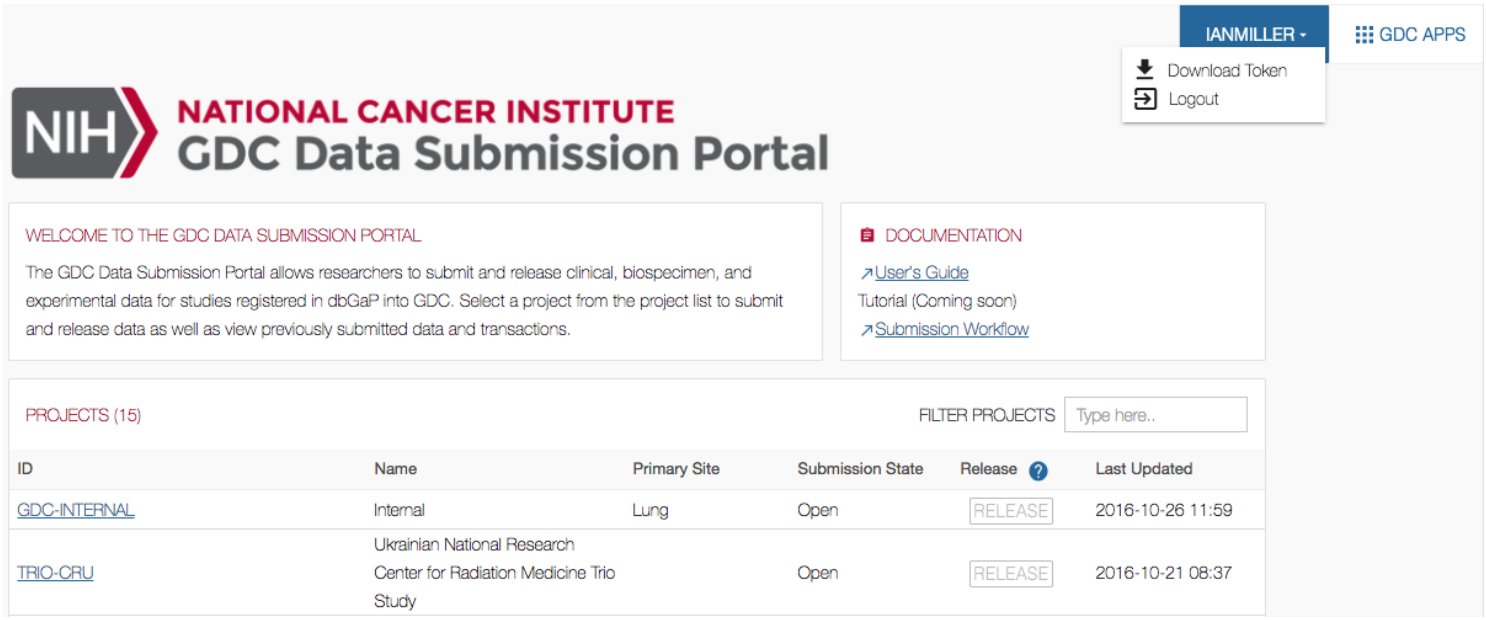
```
{
  "data": {
    "submitted_unaligned_reads": [
      {
        "file_name": "dummy.fastq",
        "id": "616eab2f-791a-4641-8cd6-ee195a10a201",
        "project_id": "GDC-INTERNAL",
        "submitter_id": "Blood-00001-aliquot_lane1_barcode23.fastq"
      }
    ]
  }
```

Obtaining an Authentication Token for Data Uploads

While biospecimen and clinical metadata may be uploaded via the GDC Data Submission Portal, file upload must be done using the Data Transfer Tool or API. An authentication token is required for data upload and can be generated on the GDC Data Submission Portal.

To generate a token, first log in to the GDC Data Submission Portal by clicking the *Login* button in the top right corner of the page. This will create a popup window that will redirect to the eRA Commons login page. After successful authentication, the GDC Submission Portal will display the username in place of the *Login* button. Here, the user Ian Miller is logged in to the GDC Submission Portal, indicated by the username IANMILLER.

Clicking the username will open a drop-down menu. Select *Download Token* from the menu to generate an authentication token.


IANMILLER
GDC APPS
Download Token
Logout
NIH
NATIONAL CANCER INSTITUTE
GDC Data Submission Portal
WELCOME TO THE GDC DATA SUBMISSION PORTAL
The GDC Data Submission Portal allows researchers to submit and release clinical, biospecimen, and experimental data for studies registered in dbGaP into GDC. Select a project from the project list to submit and release data as well as view previously submitted data and transactions.
DOCUMENTATION
User's Guide
Tutorial (Coming soon)
Submission Workflow
PROJECTS (15)
FILTER PROJECTS
Type here..
ID
Name
Primary Site
Submission State
Release
Last Updated
GDC-INTERNAL
Internal
Lung
Open
RELEASE
2016-10-26 11:59
TRIO-CRU
Ukrainian National Research Center for Radiation Medicine Trio Study
Open
RELEASE
2016-10-21 08:37

Chapter 4

Data Download and Upload

Data Downloads and Uploads

Downloads

Downloading Data Using a Manifest File

A convenient way to download multiple files from the GDC is to use a manifest file generated by the GDC Data Portal. After generating a manifest file (see Preparing for Data Download and Upload for instructions), initiate the download using the GDC Data Transfer Tool by supplying the **-m** or –**manifest** option, followed by the location and name of the manifest file. OS X users can drag and drop the manifest file into Terminal to provide its location.

The following is an example of a command for downloading files from GDC using a manifest file:

```
gdc-client download -m  /Users/JohnDoe/Downloads/gdc_manifest_6746fe840d924cf623b4634b5ec6c630bd4c06b5.txt
```

Downloading Data Using GDC File UUIDs

The GDC Data Transfer Tool also supports downloading of one or more individual files using UUID(s) instead of a manifest file. To do this, enter the UUID(s) after the download command:

```
gdc-client download 22a29915-6712-4f7a-8dba-985ae9a1f005
```

Multiple UUIDs can be specified, separated by a space:

```
gdc-client download e5976406-473a-4fbd-8c97-e95187cdc1bd fb3e261b-92ac-4027-b4d9-eb971a92a4c3
```

Resuming a Failed Download

The GDC Data Transfer Tool supports resumption of interrupted downloads. To resume an incomplete download, repeat the download of the manifest or UUID(s) in the same folder as the initial download. Failed downloads will appear in the destination folder with a .partial extension. This feature allows users the ability to identify quickly where the download stopped. For large downloads this feature can let the user identify where the download was interrupted and edit the manifest accordingly.

```
gdc-client download f80ec672-d00f-42d5-b5ae-c7e06bc39da1
```

Downloading Controlled-Access Data

A user authentication token is required for downloading Controlled-Access Data from GDC. Tokens can be obtained from the GDC Data Portal (see instructions in Obtaining an Authentication Token). Once downloaded, the token *file* can be passed to the GDC Data Transfer Tool using the **-t** or –**token-file** option:

```
gdc-client download -m gdc_manifest_e24fac38d3b19f67facb74d3efa746e08b0c82c2.txt -t
    gdc-user-token.2015-06-17T09-10-02-04-00.txt
```

Directory structure of downloaded files

The directory in which the files are downloaded will include folders named by the file UUID. Inside these folders, along with the the data and zipped metadata or index files, will exist a logs folder. The logs folder contains state files that insure that downloads are accurate and allow for resumption of failed or prematurely stopped downloads. While a download is in progress a file will have a .partial extension. This will also remain if a download failed. Once a file is finished downloading the extension will be removed. If an identical manifest is retried another attempt will be made to download files containing a .partial extension.

```
C501.TCGA-BI-A0VR-10A-01D-A10S-08.5_gdc_realn.bam.partial  logs
```

Uploads

Uploading Data Using a Manifest File

GDC Data Transfer Tool supports uploading molecular data using a manifest file to the Data Submission Portal. The manifest file for submittable data files can be retrieved from the GDC Data Submission Portal, or directly from the GDC Submission API given a submittable data file UUID. The user authentication token file needs to be specified using the **-t** or –**token-file** option.

First, generate an upload manifest, either using the GDC Data Submission Portal, or using a call to the GDC Submission API `manifest` endpoint (as in the following example):

```
export
    token=ALPHANUMERICTOKEN-01234567890+AlPhAnUmErIcToKeN=0123456789-ALPHANUMERICTOKEN-01234567890+AlPhAnUmErIcT

curl --header "X-Auth-Token: $token"
    'https://api.gdc.cancer.gov/submission/CGCI/BLGSP/manifest?ids=460ad2fe-5a7f-4797-9e18-336d33e21444'
    >manifest.yml
```

```
gdc-client upload --manifest manifest.yml --token-file token.txt
```

Uploading Data Using a GDC File UUID

The GDC Data Transfer Tool also supports uploading molecular data using a file UUID. The tool will first make a request to get the filename and project id from GDC API, and then upload the corresponding file from the current directory.

```
gdc-client upload cd939bdd-b607-4dd4-87a6-fad12893932d -t token.txt
```

Resuming a Failed Upload

By default, GDC Data Transfer Tool uses multipart transfer to upload files. If an upload failed but some parts were transmitted successfully, a resume file will be saved with the filename *resume_[manifest_filename]*. Running the upload command again will resume the transfer of only those parts of the file that failed to upload in the previous attempt.

```
gdc-client upload -m manifest.yml -t token
```

Deleting Previously Uploaded Data

Previously uploaded data can be replaced with new data by deleting it first using the –**delete** switch:

```
gdc-client upload -m manifest.yml -t token --delete
```

Recurrent Transfers of Very Large Datasets over High-speed Networks

Institutions that regularly transfer very large volumes of data between GDC facilities (located in Chicago, IL, USA) and a geographically remote location over gigabit+ networks may benefit from using the UDT mode of the GDC Data Transfer Tool. **UDT mode** is an advanced feature that uses UDT, or User Datagram Protocol (UDP)-based Data Transfer, instead of the ubiquitous Transmission Control Protocol (TCP) protocol. Please if you are interested in learning more about this feature.

Troubleshooting

Invalid Token

An error message about an 'invalid token' means that a new authentication token needs to be obtained from the GDC Data Portal or the GDC Data Submission Portal as described in Preparing for Data Download and Upload.

```
403 Client Error: FORBIDDEN: {
     "message": "Your token is invalid or expired, please get a new token from GDC Data Portal"
     }
```

dbGaP Permissions Error

Users may see the following error message when attempting to download a file from GDC:

```
403 Client Error: FORBIDDEN: {
     "message": "You don't have access to the data: Please specify a X-Auth-Token"
     }
```

This error message indicates that the user does not have dbGaP access to the project to which the file belongs. Instructions for requesting access from dbGaP can be found here.

File Availability Error

Users may also see the following error message when attempting to download a file from GDC:

```
403 Client Error: FORBIDDEN: {
   "message": "You don't have access to the data: Requested file abd28349-92cd-48a3-863a-007a218de80f
       does not allow read access"
   }
```

This error message means that the file is not available for download. This may be because the file has not been uploaded or released yet or that it is not a file entity.

GDC Upload Privileges Error

Users may see the following error message when attempting to upload a file:

```
Can't upload: {
    "message": "You don't have access to the data: You don't have create role to do 'upload'"
    }
```

This means that the user has dbGaP read access to the data, but does not have GDC upload privileges. Users can contact The database of Genotypes and Phenotypes (dbGaP) to request upload privileges.

File in Uploaded State Error

Re-uploading a file may return the following error:

```
Can't upload: {
     "message": "File in uploaded state, upload not allowed"
     }
```

To resolve this issue, delete the file using the **–delete** switch before re-uploading.

Microsoft Windows Executable Error

Attempting to run gdc-client.exe by double-clicking it in the Windows Explorer will produce a window that blinks once and disappears.

This is normal, the executable must be run using the command prompt. Click 'Start', followed by 'Run' and type 'cmd' into the text bar. Then navigate to the path containing the executable using the 'cd' command.

Chapter 5

Key Terms

The following table provides definitions and explanations for terms and acronyms relevant to the content presented within this document.

Term	Definition
eRA	Electronic Research Administration
GDC	Genomic Data Commons
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
ID	Identifier
NCI	National Cancer Institute
TCGA	The Cancer Genome Atlas
TCP	Transmission Control Protocol
UUID	Universally Unique Identifier

Chapter 6

Release Notes

Data Transfer Tool Release Notes

v1.2.0

- **GDC Product**: Data Transfer Tool
- **Release Date**: October 31, 2016

New Features and Changes

- Better handling of connectivity interruptions

Bugs Fixed Since Last Release

- Uploads via manifest file has been fixed.
- Legacy -i/–identifier flag removed.
- Improved error messaging when uploading without a token.

Known Issues and Workarounds

- Use of non-ASCII characters in token passed to Data Transfer Tool will produce incorrect error message “Internal server error: Auth service temporarily unavailable”.
- On some terminals, dragging and dropping a file into the interactive client will add single quotes (‘’) around the file path. This causes the interactive client to misinterpret the file path and generate an error when attempting to load a manifest file or token.
  - *Workaround:* Manually type out the file name or remove the single quotes from around the file path.
- When any files mentioned in the upload manifest are not present in the upload directory the submission will hang at the missing file.
  - *Workaround:* Edit the manifest to specify only the the files that are present in the upload directory for submission or copy the missing files into the upload directory.
- Upload flags –path/-f do not modify the upload path as expected.
  - *Workaround:* Copy the Data Transfer Tool into the the root of the submittable data directory and run from there.
- Submission manifest field **local_file_path:** does not modify upload path expected.
  - *Workaround:* Run Data Transfer Tool from root of the submittable data directory so that data is in the current working directory of the Data Transfer Tool.

v1.1.0

- **GDC Product**: Data Transfer Tool
- **Release Date**: September 7, 2016

New Features and Changes

- Partial extension added to all download files created during download. Removed after successful download.
- Number of processes started by default changed to 8 (-n flag).

Bugs Fixed Since Last Release

- None to report.

Known Issues and Workarounds

- Use of non-ASCII characters in token passed to Data Transfer Tool will produce incorrect error message "Internal server error: Auth service temporarily unavailable".
- On some terminals, dragging and dropping a file into the interactive client will add single quotes (') around the file path. This causes the interactive client to misinterpret the file path and generate an error when attempting to load a manifest file or token.
  - *Workaround:* Manually type out the file name or remove the single quotes from around the file path.
- Use of a manifest file for uploads to the Submission Portal will produce an error message "ERROR: global name 'read_manifest' is not defined".
  - *Workaround:* Upload files via UUID instead or use the API/Submission Portal.

v1.0.1

- **GDC Product**: Data Transfer Tool
- **Release Date**: June 2, 2016

New Features and Changes

- MD5 checksum verification of downloaded files.
- BAM index files (.bai) are now automatically downloaded with parent BAM.
- UDT mode included to help improve certain high-speed transfers between the GDC and distant locations.

Bugs Fixed Since Last Release

- None to report.

Known Issues and Workarounds

- Use of non-ASCII characters in token passed to Data Transfer Tool will produce incorrect error message "Internal server error: Auth service temporarily unavailable".
- On some terminals, dragging and dropping a file into the interactive client will add single quotes (') around the file path. This causes the interactive client to misinterpret the file path and generate an error when attempting to load a manifest file or token.
  - *Workaround:* Manually type out the file name or remove the single quotes from around the file path.

v1.0.0

- **GDC Product**: Data Transfer Tool
- **Release Date**: May 26, 2016

New Features and Changes

- Single-thread and multi-threaded download capability
- User-friendly command line interface
- Progress bars provide visual representation of transfer status
- Optional interactive (REPL) mode
- Detailed help menus for upload and download functionality
- Support for authentication using a token file
- Support for authentication using a token string
- Resumption of incomplete uploads and downloads
- Initiation of transfers using manifests
- Initiation of transfers using file UUIDs
- Advanced configuration options
- Binary distributions available for Linux (Ubuntu), OS X, and Windows

Bugs Fixed Since Last Release

- None to report.

Known Issues and Workarounds

- Use of non-ASCII characters in token passed to Data Transfer Tool will produce incorrect error message "Internal server error: Auth service temporarily unavailable".
- On some terminals, dragging and dropping a file into the interactive client will add single quotes (') around the file path. This causes the interactive client to misinterpret the file path and generate an error when attempting to load a manifest file or token.
  - *Workaround:* Manually type out the file name or remove the single quotes from around the file path.