1 Change Log 2

2 Introduction 3
   2.1 Organizational Entities ............................................. 3
   2.2 Identifiers ................................................................. 3
   2.3 System Workflow .......................................................... 4

3 Submission Process 5
   3.1 States ............................................................................ 6
   3.2 User Authentication .......................................................... 7
   3.3 Submission Steps ............................................................. 7

4 Software Installation 10
   4.1 Check Installation ............................................................ 10

5 Network/Firewall Considerations 12

6 Data Preparation 13
   6.1 Data Files ....................................................................... 13
   6.2 Metadata ......................................................................... 14
   6.3 gtupload ........................................................................ 21
   6.4 GeneTorrent .................................................................... 22
   6.5 cgquery .......................................................................... 23
   6.6 Validation Web Services ...................................................... 23
Revision History
### CHANGE LOG

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0.6</td>
<td>03/14/2014</td>
<td>Added Updates to Previously Submitted Metadata section.</td>
</tr>
<tr>
<td>3.0.5</td>
<td>03/04/2014</td>
<td>Rewrote Software section to refer to User Guide and just list the differences, updated for GeneTorrent 3.8.5</td>
</tr>
<tr>
<td>3.0.4</td>
<td>01/30/2014</td>
<td>Updated the state table to define redaction in section 3.1</td>
</tr>
<tr>
<td>3.0.3</td>
<td>09/26/2013</td>
<td>Changed frozen sample preservation to FROZEN</td>
</tr>
<tr>
<td>3.0.2</td>
<td>09/24/2013</td>
<td>Add FASTQ as a data file type supported for upload, fixes cgsubmit vs. gtupload mixup, updated introduction</td>
</tr>
<tr>
<td>3.0.1</td>
<td>09/23/2013</td>
<td>Describe markup of sample preservation (FFPE, frozen), minor fixes</td>
</tr>
</tbody>
</table>
INTRODUCTION

This document describes the submission protocol for uploading genomic sequencing information to the CGHub repository for the Cancer Genome Atlas Project (TCGA), a National Institutes of Health (NIH) Roadmap study sponsored by the National Cancer Institute (NCI) and National Human Genome Research Institute (NHGRI).

For the TCGA study, sequencing and alignment data comes from human clinical samples, which are considered personally identifying information (PII). In order to support research use guidelines and enforce patient privacy rights, only NCI authorized users of TCGA can access the genomic data via protected CGHub Web Services. The meta-data describing the sequence, captured as XML-based metadata descriptors, is available to anyone through the public CGHub Web Services.

Submitters of data to TCGA need to follow similar security procedures for submission through the protected CGHub Web Services. Authorized personnel from the Genome Sequencing Centers will be provided accounts to enable CGHub submission. This account is separate from NCI study authorization, providing upload only access to CGHub.

This document is based on guidelines originally developed by NCI, NCBI and the TCGA study researchers. While the spirit of the NCBI SRA repository is embodied in the CGHub submission process, it is different and unique enough that it will require some adjustments by the user community to submit and access the CGHub repository.

The TCGA project is the first study to utilize the CGHub repository; in addition the Therapeutically Applicable Research to Generate Effective Treatments (TARGET) and the Cancer Genome Characterization Initiative (CGCI) studies are/will be stored in CGHub.

2.1 Organizational Entities

https://cghub.ucsc.edu/docs/user/overview.html#organizational-entities

2.2 Identifiers

https://cghub.ucsc.edu/docs/user/query.html#identifiers
2.3 System Workflow

https://cghub.ucsc.edu/docs/user/overview.html#system-workflow
The TCGA has mandated submission of primary sequencing data in the form of binary sequence Alignment/Mapping (BAM) as well as FASTQ files for mRNA-seq experiments. The payload of the BAM file contains both the sequencing data (in bases, quality scores, and read names produced by the instrument) and read placements on a specific human assembly reference with annotations about strand, alignment, and quality features. The FASTQ files simply store the read sequences with the associated sequencer qualities per base. The mRNA-seq FASTQ files are provided in addition to their BAM equivalents for groups who simply want just the raw sequence data.

All data submissions must include submission metadata compliant with the SRA 1.5 XML schema. This schema was developed and maintained by NCBI and the International Nucleotide Sequence Database Collaboration (INSDC, http://www.insdc.org/).

Users must execute the following steps to make a new data submission to CGHub:

1. Create and save a user authorization key using the CGHub authorization web page. Each submitter has to have an authorized account with CGHub prior to submission.

2. Submit the metadata to CGHub using the cgsubmit script. This step will perform all necessary validation to ensure that the metadata are complete and correctly formatted. All validation errors will be returned immediately so that they can be corrected and resubmitted. A submission will not be accepted until the metadata completely validates. Validation rules are a work in progress but are generally decided in consultation with the TCGA Program Office (PO) and the submitting centers.

3. Submit the data files to CGHub using the GeneTorrent gtupload client. The GeneTorrent gtupload program will complete after successful transfer of the sequence data to CGHub. In the case of an upload interruption, the gtupload program should be able to be restarted to resume an upload from the point of interruption in the general case. Uploaders are encouraged to contact CGHub support (support@cghub.ucsc.edu) at any time in this process should there appear to be a problem.

4. Monitor the submission status using the cgquery tool (or directly access the CGHub Web Services) to determine when the upload and validation processing is complete.

---

1 The SRA 1.5 Schema is defined by a number of XML Schema Definition files (XSDs) here: http://www.ncbi.nlm.nih.gov/viewvc/v1/trunk/sra/doc/SRA_1-5/

The SRA 1.5 release notes can be found at http://www.ncbi.nlm.nih.gov/viewvc/v1/trunk/sra/doc/SRA_1-5c/XML_1_5_Specification_BK.pdf?view=log
3.1 States

The following are the list of possible states that an uploaded entry will go through in the life cycle of a CGHub entry:

1. “submitted” Metadata has been successfully validated and submitted to CGHub, GeneTorrent upload has not started or reached the point of updating state

2. “uploading” GeneTorrent has been started, a GTO file has been created, data file(s) are either about to be transferred or are transferring, or prematurely terminated

3. “validating_data” GeneTorrent has finished transferring successfully, checksums are being run on data files(s), and matched against submitted metadata

4. “augmenting_data” All uploading and validation steps have been performed successfully and the data file is being processed to produce 0 or more ancillary data files (such as an index for a BAM file). If the augmenting fails the state will be changed to bad_data.

5. “bad_data” If validating_data or augmenting_data fails, the object state will go to the bad_data sink state, manual intervention is required to recover from this state. Usually an explanatory message is put in the reason field of the metadata.

6. “validating_sample” Once the data file(s) have been validated successfully, sample/aliquot level metadata is retrieved from the TCGA DCC web services and linked to the uploaded entry’s analysis UUID via the Aliquot UUID. Note, if there is a problem with the DCC’s web services, or the sample level metadata cannot be found, the state will stay in validating_sample and the upload manager (SPM) will frequently automatically retry the query against the TCGA DCC’s web services until either the DCC’s services return correct data or there is manual intervention. As in the bad_data state, in case of an error an explanatory message is put in the reason field of the metadata.

7. “suppressed” Currently, on request of the submitting center (e.g. GSC) entries will be suppressed. In a suppressed state, the metadata can still be queried but associated data file(s) cannot be downloaded.

Note: For Submitters: Suppressions take precedence over redactions (below) and are generally not temporary. The reason field for a suppressed analysis_id will not be overwritten by a later redaction of the same analysis_id.

8. “redacted” The Program Office for the project (TCGA, TARGET, or CGCI) decides that a participant (patient/case) has sample problems (e.g. duplication) or consent issues and requests that all analysis files (e.g. BAMs, FASTQs) associated with the participant are removed from being available for download (removed from the “live” category). This can happen directly or via coordination with the TCGA Data Coordinating Center (DCC). Redactions can be temporary.

9. “live” If an entry reaches this state it signifies the following:
   (a) data file(s) checksums match metadata
   (b) sample information from TCGA DCC was successfully retrieved for this sample/aliquot
3.2 User Authentication

Personnel of authorized sequencing centers that submit sequences to CGHub must obtain an authorization credential based on an eRA username and password. The credential allows the submitter to make multiple submissions over a period of time as well as run batch submission jobs without needing to reveal their password in the job script.

Once the CGHub credential file has been obtained, the eRA username, used to create the credential file, must be registered with CGHub as an authorized uploader account. Contact support@cghub ucsc.edu with the username to request uploader authorization.

A submitter can authenticate and retrieve an authorization credential using the following steps:

1. Enter the following URL into a Web browser: https://cghub.ucsc.edu/secure
2. The browser will be redirected to the NIH eRA login page via the CGHub Shibboleth middleware system; the user authentication screen prompts for an eRA username and password.
3. If successfully authenticated, the user will be prompted to save the credential (key file) to a local file name. This credential is valid for 365 days and should be saved in a secure (restricted) location.
4. Request upload authorization from support@cghub.ucsc.edu
5. The submitter passes the credential to the CGHub submission tools (cgsubmit and gtupload) for authorization.

When the credential expires, or the user’s account changes, the submitter can return to the CGHub web page to generate a new credential: https://cghub.ucsc.edu/secure

3.3 Submission Steps

The submitting center (e.g. GSC) is responsible for producing sequence data files and associated metadata from a specific biospecimen using high-throughput, next generation sequencing machines and alignment/mapping software.

The following sections list the steps required for the submitting center personnel to submit a new Analysis Object (sequence data and metadata) to CGHub.

Note: The authorization token created in the previous section is used in some of these steps.

3.3.1 Create an Analysis Object UUID

Run the uuidgen program to create a universally unique identifier (UUID) to associate with the analysis object. The Analysis Object UUID ensures that this submission can be referenced in future accesses and is
unique across the entire TCGA submissions community. The generated UUID complies with the RFC4122 specification.

### 3.3.2 Create a directory for associating Analysis Object data files

All Analysis Object-related files (data and metadata) need to be placed in the same directory. The directory is named using the Analysis Object UUID.

### 3.3.3 Copy the sequence data and metadata to the Analysis Object directory

Make a copy of all data in the Analysis Object directory. User can either make local copies or else create symlinks to where ever the data is originally stored using the Linux: `ln -s {target-filename} {symbolic-filename}` command.

The directory must contain `analysis.xml`, `experiment.xml`, `run.xml` and the one BAM or FASTQ file referenced within the `analysis.xml` FILES element.

**NOTE:** In the case of a FASTQ file, the accepted practice is the following:

1. split the set of reads into 2 FASTQ files with one file containing the first read mate in all the pairs and the second set containing all the second read mates
2. package the two read FASTQ files into a GZipped TAR archive with extension `.tar.gz`
3. the output file of step 2 should be file references in `analysis.xml` with its MD5 checksum

### 3.3.4 Submit the Analysis Object to CGHub for validation

Run the `cgsubmit` python script to request CGHub to validate the XML and write them to the CGHub metadata database. The XML files will be submitted to CGHub using a HTTPS POST on the [https://cghub.ucsc.edu/cghub/metadata/analysis/submit](https://cghub.ucsc.edu/cghub/metadata/analysis/submit) Web Service.

CGHub will validate the structural metadata required to identify and manage the sequencing data. It will also perform any project specific validation rules such as adherence to a common vocabulary.

These rules are defined by the Project Office and may vary between projects.

If successful, the metadata client will generate a `manifest.xml` file that can be passed to the `gtupload` client to upload the actual data files (e.g. BAMs or FASTQs). Users can retrieve the “`analysis_id`” (UUID) from this file, or later query the CGHub [https://cghub.ucsc.edu/cghub/metadata/analysisFull](https://cghub.ucsc.edu/cghub/metadata/analysisFull) REST Web Service Interface (WSI).

In case the manifest file is removed accidentally but is still needed for future uploading or for record keeping purposes, the manifest file can be re-generated without re-submitting by using the `cgsubmit` command with the option `--validate-only` with the same submitted directory.

In the event that errors are found in the submission, an error log will be returned with descriptive errors to help isolate the failures.

If desired, `cgsubmit --validate-only` can be called repeatedly to validate that the submission metadata is complete and accurate prior to actual submission. This allows the submitter to iteratively locate and
correct errors in the submission ahead of time. This operation is fully automatic at CGHub and will return results almost immediately.

3.3.5 Transfer the Analysis Object data to CGHub

Once the metadata has been submitted, the user passes the manifest.xml file to the CGHub `gtupload` client to initiate data transfer. `gtupload` then segments the file and begins parallel file transfer sessions of the file pieces over SSL. The GeneTorrent protocol will manage transmissions errors on any of the file pieces and will reassemble the file on CGHub.

Once the transfer is complete, CGHub will perform a series of validation steps prior to making the data available for download. CGHub will also automatically compute the MD5 checksum and compare against the value in the submission XML files. CGHub and the DCC Metadata Web Services will be queried to determine if the sample is valid and is in an active state (e.g. has not been redacted).

- If the sample cannot be found, the state will be set to “validating_sample”.
- If the sample is found, but has been redacted, the state will be set to “suppressed”.

In both cases, CGHub will periodically poll the DCC to see if the state has changed.

GeneTorrent will optionally calculate an MD5 checksum for each file and validate that it matches the MD5 checksum in the manifest XML to confirm that it has the right files before transferring the data.

Once all of the validation steps succeed, the submission state becomes live and the Analysis Object is available for download. CGHub will continue to periodically poll the DCC for all live samples for a change in their redaction status and will suppress all Analysis Objects associated with samples that have been redacted.

3.3.6 Check state of Analysis Object

Users can query the state of a submission at any time using the `cgquery` python script or directly through the CGHub `https://cghub.ucsc.edu/cghub/metadata/analysisFull` REST Web Services. An example query would be using the analysis ID to query for a specific Analysis Object: `https://cghub.ucsc.edu/cghub/metadata/analysisFull?analysis_id=3a8e6a74-137f-468f-8987-fa0acdde2836`
SOFTWARE INSTALLATION

Submitters will use two programs when interacting with CGHub: gtupload and cgsubmit. The installation of these two programs follows a procedure virtually identical to that of gtdownload and cgquery, the programs used for downloading from CGHub. We will not duplicate these instructions here and instead have you refer to the Software section of the CGHub User Guide at https://cghub.ucsc.edu/docs/user/software.html.

To install gtupload and cgsubmit follow the instructions for gtdownload and cgquery with the following substitutions:

- Download artifacts from the CGHub Upload Client Software page at https://cghub.ucsc.edu/software/submitters.html

- If you are installing the binary .tar.gz distribution, substitute GeneTorrent-download-*.x86_64.tar.gz with GeneTorrent-upload-*.x86_64.tar.gz. You may want to install the download client alongside the upload client. In that case, extract both .tar.gz files in the same directory.

- If you are installing .rpm packages, substitute GeneTorrent-download-*.x86_64.rpm with GeneTorrent-upload-*.x86_64.rpm. You may want to install the download client alongside the upload client. In that case simply install all three .rpm packages: GeneTorrent-common-*.rpm, GeneTorrent-download-*.rpm and GeneTorrent-upload-*.rpm.

- If you are installing .deb packages, substitute genetorrent-download_*_amd64.deb with genetorrent-upload_*_amd64.deb. You may want to install the download client alongside the upload client. In that case simply install all three .deb packages: genetorrent-common_*_.deb, genetorrent-download_*_.deb and genetorrent-upload_*_.deb.

4.1 Check Installation

To verify that the installation succeeded and that the correct version of the software is installed type the following command:

gtupload --version
Expected result (the exact version, SCM reference and build number may differ):
GeneTorrent gtupload release 3.8.5 (SCM REV: git ref: 1b8e7685d6, build: 91)

For technical or operational questions 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

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: https://cghub.ucsc.edu/help/help.html
DATA PREPARATION

Centers must have a TCGA center designation in order to submit sequencing data. Current TCGA centers are:

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

6.1 Data Files

6.1.1 BAM Data Files

BAM files shall follow the requirements of BAM file submission for NIH projects. For details, please see SRA Analysis Submission Guide. BAM files should not be compressed or wrapped into another archive container.

6.1.2 FASTQ Data Files

Files formatted as FASTQ only for RNA-seq sequences are allowed to be uploaded as well. They should be split into mate-grouped FASTQ files and submitted together as a GZipped TAR archive with extension .tar.gz.

6.1.3 Supplemental Data Files

CGHub can accept additional files, such as raw read files in .srf format. These files will be uploaded stored and downloaded along with the other files referenced by the analysis object. The MD5 checksum will be
validated, but no additional validation will be performed.

6.2 Metadata

A TCGA submission to CGHub consists of three xml formatted files that provide attributes about the sequence data files. The metadata contained in the xml files must be 100% SRA 1.5 schema compliant. In addition, certain fields are required by CGHub to manage the Analysis Object in the repository. The metadata files describe a single analysis object within the analysis.xml, one or more experiments within the experiment.xml and one or more runs within the run.xml

**Note:** The SRA 1.5 schema permits analysis.xml to contain multiple analysis objects. However, CGHub stores the metadata separately for each Analysis Object so submissions must separate each analysis object. If needed, pre-processing tools can be provided that will reformat an XML with multiple analyses into the CGHub-compliant submission format.

All data must be submitted with reference to the NIH study identifier for TCGA. A biospecimen sample for each Analysis Object must exist within the TCGA DCC in order to associate this submission to the proper study for correct download user authorization. For TCGA, the aliquot UUID is used to identify the unique biospecimen.

The Analysis Object metadata will be publicly available from CGHub. Consequently, do not include identifying information in the XML metadata. If information is restricted to the library preparation and run conditions this will not be an issue.

All of the examples below depict the minimum required metadata. CGHub will reject the submission if the minimum metadata required for CGHub operation is not specified in the XML. However, the submitter may include any of the SRA 1.5 compliant fields. These will be stored, searchable, and retrievable at CGHub. Each study can specify additional validation rules for further metadata checks. These study specific rules are configurable by CGHub operations staff upon request by the study leaders.

6.2.1 Updates to Previously Submitted Metadata

CGHub supports manual updates to the submitted raw XML-based metadata to fix errors. The submitters are encouraged to send a gzipped tar archive of all such entries in CGHub identified by analysis_id (uuid) as the subdirectory with the full set of metadata xml files as the contents.

**Example:**

c676ee58-3e2c-4d56-a09e-c7893000b736/
analysis.xml
experiment.xml
run.xml

Changes to the FILES block of the raw metadata are NOT allowed due to CGHub’s policy that all file related data and metadata are immutable. In this case a new submission must be made.
6.2.2 Study

The experiment and analysis XML must reference the pre-defined TCGA study.

6.2.3 Samples

Sample metadata is maintained by the DCC and dbGaP. CGHub data and DCC data will be linked by the aliquot UUID passed in the analysis and experiment XML. The aliquot uuid is created by the BCR and provided to the DCC by the BCR in the TCGA_BIOSPECIMEN_XML information.

6.2.4 Analysis.xml

Analysis objects are the primary container for submitting and downloading sequence data. For TCGA, these represent an alignment and read data contained within a BAM file. The root element of the analysis.xml file must contain a schema declaration referencing the SRA 1.5 schema document for analysis objects:

```xml
<ANALYSIS_SET
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

Analysis.xml Structural Metadata

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>center_name</td>
<td>Name of the submitting center. See Table 1 for the list of valid center names. Center defined name for this analysis object.</td>
</tr>
<tr>
<td>alias</td>
<td>Center defined name for this analysis object.</td>
</tr>
<tr>
<td>accession</td>
<td>Empty for new submissions. Saved in CGHub for data migrated from SRA.</td>
</tr>
</tbody>
</table>

Example:

```xml
<ANALYSIS center_name="WUGSC"
alias="TCGA-BH-A0H5-01A-21D-A10Y-09_IlluminaA-DNASeq_exome"/>
```

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>refname</td>
<td>dbGaP accession (For TCGA, this is phs000178)</td>
</tr>
<tr>
<td>refcenter</td>
<td>NHGRI (always)</td>
</tr>
<tr>
<td>accession</td>
<td>For TCGA this is SRP000677</td>
</tr>
</tbody>
</table>
Example:

<STUDY_REF refcenter="NHGRI" refname="phs000178" accession="SRP000667"/>

Table 6.3: analysis.xml <TARGET>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sra_object_type</td>
<td>SAMPLE</td>
</tr>
<tr>
<td>refname</td>
<td>aliquot UUID managed by the DCC. String must be a valid RFC 4122</td>
</tr>
<tr>
<td></td>
<td>UUID (e.g. nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn)</td>
</tr>
<tr>
<td>accession</td>
<td>Empty for new submissions. Saved in CGHub for data migrated from SRA.</td>
</tr>
</tbody>
</table>

Example:

<TARGET sra_object_type="SAMPLE" refcenter="TCGA"
        refname="1592ccef-f2b8-48f5-b61d-9c467b608671"/>

Table 6.4: analysis.xml <DATA_BLOCK/FILES>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>filename</td>
<td>The name of the file containing the run data.</td>
</tr>
<tr>
<td>filetype</td>
<td>Type of file. For TCGA this must be bam.</td>
</tr>
<tr>
<td>checksum</td>
<td>32 character hexadecimal checksum string.</td>
</tr>
<tr>
<td>checksum_method</td>
<td>Method used to calculate checksum. For TCGA, this must be MD5.</td>
</tr>
</tbody>
</table>

For TCGA, analysis objects will point to a single BAM file.

Example:

<DATA_BLOCK>
  <FILES>
    <FILE checksum="eb0cd5098783397824d085f35de5de03"
            checksum_method="MD5" filetype="bam"
            filename="UNCID_258704.TCGA-A5-A0G5-01A-11R-A040-07.101011_UNC8-
            RDR3001640_00037_FC_62M6WAAXX.2.trimmed.annotated.translated_to_genomic.bam"/>
  </FILES>
</DATA_BLOCK>

XML Document: analysis.xml

XML Element: ANALSYS_TYPE=REFERENCE_ALIGNMENT

Example:

<ANALSYS_TYPE>
  <REFERENCE_ALIGNMENT>

6.2. Metadata
Table 6.5: analysis.xml <REFERENCE_ALIGNMENT/RUN_LABELS/RUN>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>refcenter</td>
<td>References the alias of the run object used by this analysis.</td>
</tr>
<tr>
<td>refname</td>
<td>References the alias of the run object used by this analysis.</td>
</tr>
<tr>
<td>read_group_label</td>
<td>32 character hexadecimal checksum string.</td>
</tr>
<tr>
<td>checksum_method</td>
<td>Method used to calculate checksum. For TCGA, this must be MD5.</td>
</tr>
<tr>
<td>data_block_name</td>
<td>Strings may contain alpha-numeric characters</td>
</tr>
<tr>
<td>seq_label</td>
<td>Strings may contain alpha-numeric characters</td>
</tr>
<tr>
<td>accession</td>
<td>Empty for new submissions. Saved in CGHub for data migrated from SRA.</td>
</tr>
</tbody>
</table>

Example:

```
<REFERENCE_ALIGNMENT>
  <RUN_LABELS>
    <RUN accession="SRR327242" refcenter="BI"
      refname="C01PRACXX110628.1.TTGTCTAT.C484.TCGA-06-5412-10A-01D-1696-08.2.bam"
      read_group_label="C01PR.1"
      data_block_name="C01PRACXX"/>
    <RUN accession="SRR327243"
      refcenter="BI"
      refname="C01PRACXX110628.2.TTGTCTAT.C484.TCGA-06-5412-10A-01D-1696-08.2.bam"
      read_group_label="C01PR.2" data_block_name="C01PRACXX"/>
  </RUN_LABELS>
</REFERENCE_ALIGNMENT>
```

Analysis.xml Project Metadata

CGHub uses configurable validation rules defined by the Project Office and user community. The following parameters are required for the TCGA project. Please see the TCGA Metadata mind map for the current working discussion:

http://www.mindmeister.com/118955602

Table 6.6: analysis.xml <REFERENCE_ALIGNMENT/SEQ_LABELS/SEQUENCE>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq_label</td>
<td>Strings may contain alpha-numeric characters, TODO</td>
</tr>
</tbody>
</table>

Example:

```
<REFERENCE_ALIGNMENT>
  <SEQ_LABELS>
    <SEQUENCE seq_label="chr7"
      data_block_name="C484.TCGA-06-5412-10A-01D-1696-08.2"
      gi="89161213"
      accession="NC_000007.12"/>
    <SEQUENCE seq_label="chr2"
      data_block_name="C484.TCGA-06-5412-10A-01D-1696-08.2"
      gi="89161199"
      accession="NC_000002.10"/>
  </SEQ_LABELS>
</REFERENCE_ALIGNMENT>
```
6.2.5 Experiment.xml

Experiment objects represent the library used for sequencing. The root element of the experiment.xml file must contain a schema declaration referencing the SRA 1.5 schema document for experiment objects:

```xml
<EXPERIMENT_SET xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

### Experiment.xml Structural Metadata

#### Table 6.7: experiment.xml <EXPERIMENT>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>center_name</td>
<td>Name of the submitting center. See the Table 1 for list of valid center names.</td>
</tr>
<tr>
<td>alias</td>
<td>Center defined name for this experiment object.</td>
</tr>
<tr>
<td>accession</td>
<td>Empty for new submissions. Saved in CGHub for data migrated from SRA.</td>
</tr>
</tbody>
</table>

Example:

```xml
<EXPERIMENT center_name="Bi" alias="7290.WR20892.Solexa-19331.61C25AAAXX100218.P">
```

#### Table 6.8: experiment.xml <STUDY_REF>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>center_name</td>
<td>Name of the submitting center. See the Table 1 for list of valid center names.</td>
</tr>
<tr>
<td>refname</td>
<td>dbGaP accession (For TCGA, this is phs000178)</td>
</tr>
<tr>
<td>refcenter</td>
<td>NHGRI (This is the value used in SRP000677)</td>
</tr>
<tr>
<td>accession</td>
<td>For TCGA this is SRP000677.</td>
</tr>
</tbody>
</table>

Example:

```xml
<STUDY_REF refcenter="NHGRI" refname="phs000178" accession="SRP000667"/>
```

#### Table 6.9: experiment.xml </DESIGN/SAMPLE_DESCRIPTOR>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>refname</td>
<td>aliquot UUID defined in the DCC. String must be a valid RFC 4122 UUID (e.g. nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnn )</td>
</tr>
<tr>
<td>refname</td>
<td>dbGaP accession (For TCGA, this is phs000178)</td>
</tr>
<tr>
<td>refcenter</td>
<td>TCGA</td>
</tr>
<tr>
<td>accession</td>
<td>Empty for new submissions.</td>
</tr>
</tbody>
</table>

Saved in CGHub for data migrated from SRA.

Example:
Experiment.xml Project Metadata

CGHub uses configurable validation rules defined by the Project Office and user community. The following parameters are required for the TCGA project. Please see the TCGA Metadata mind map for the current working discussion: http://www.mindmeister.com/118955602

Table 6.10: experiment.xml <DESIGN/LIBRARY_DESCRIPTOR>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>library_name</td>
<td>User defined string.</td>
</tr>
<tr>
<td>library_strategy</td>
<td>WGS</td>
</tr>
</tbody>
</table>

Example:

<LIBRARY_DESCRIPTOR>
  <LIBRARY_NAME>Solexa-34237</LIBRARY_NAME>
  <LIBRARY_STRATEGY>WXS</LIBRARY_STRATEGY>
  <LIBRARY_SOURCE>GENOMIC</LIBRARY_SOURCE>
  <LIBRARY_SELECTION>Hybrid Selection</LIBRARY_SELECTION>
</LIBRARY_DESCRIPTOR>

Table 6.11: experiment.xml <PLATFORM> :header-rows: 1 :stub-columns: 1

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILLUMINA</td>
</tr>
</tbody>
</table>

Example:

<PLATFORM>
  <ILLUMINA>
    <INSTRUMENT_MODEL>IlluminaHiSeq 2000</INSTRUMENT_MODEL>
    <CYCLE_COUNT>152</CYCLE_COUNT>
    <SEQUENCE_LENGTH>152</SEQUENCE_LENGTH>
  </ILLUMINA>
</PLATFORM>
Experiment.xml Sample Preservation

The SRA schema currently does not provide a means to describe the method used for preserving tissue samples intended for sequencing. As a stop-gap, CGHub recommends that the GSCs include the experiment attribute `SAMPLE_PRESERVATION` in their submissions. The CGHub submission validator treats the attribute as optional, but if it is present its value must be either `FROZEN` or `FFPE`.

Example:

```xml
<EXPERIMENT_ATTRIBUTES>
  <EXPERIMENT_ATTRIBUTE>
    <TAG>SAMPLE_PRESERVATION</TAG>
    <VALUE>FFPE</VALUE>
  </EXPERIMENT_ATTRIBUTE>
</EXPERIMENT_ATTRIBUTES>
```

6.2.6 Run.xml

Run objects describe the sequencing run and its read_groups. The root element of the run.xml file must contain a schema declaration referencing the SRA 1.5 schema document for run objects:

Example:

```xml
<RUN_SET xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

Run.xml Structural Metadata

Table 6.12: run.xml `<RUN>`

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>center_name</td>
<td>Name of the submitting center. See the Table 1 for list of valid center names</td>
</tr>
<tr>
<td>alias</td>
<td>Center defined name for this run object.</td>
</tr>
<tr>
<td>accession</td>
<td>Empty for new submissions. Saved in CGHub for data migrated from SRA.</td>
</tr>
</tbody>
</table>

Example:

```xml
<RUN center_name="UNC-LCCC" alias="UNCID:31111" run_center="UNC-LCCC">```

Table 6.13: run.xml `<EXPERIMENT_REF>`

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>refcenter</td>
<td>Name of the submitting center. See the Table 1 for list of valid center names</td>
</tr>
<tr>
<td>refname</td>
<td>References the alias of the experiment object used by this run.</td>
</tr>
<tr>
<td>accession</td>
<td>Empty for new submissions. Saved in CGHub for data migrated from SRA.</td>
</tr>
</tbody>
</table>

Example:
Table 6.14: run.xml <DATA_BLOCK/FILES>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>filename</td>
<td>The name of the file containing the run data</td>
</tr>
<tr>
<td>filetype</td>
<td>Type of file. For TCGA this is typically bam. srf for RNA-Seq data files.</td>
</tr>
<tr>
<td>checksum</td>
<td>32 character hexadecimal checksum string.</td>
</tr>
<tr>
<td>checksum_method</td>
<td>Method used to calculate checksum (typically MD5).</td>
</tr>
</tbody>
</table>

For TCGA, the run file block is optional since all data is generally submitted within the BAM file referenced by the analysis object.

Any files that are referenced within the run FILES block will be uploaded to and downloaded from CGHub along with the analysis FILES.

Example:

```xml
<DATA_BLOCK>
<FILES>
  <FILE checksum="7fecc664ee312459b2cd5e7c27432db0" checksum_method="MD5" filetype="srf"
   filename="UNCID_35387.TCGA-AP-A052-01A-11R-A00V-07.100903_UNC3-DR300156_00028_FC_62ELKAAXX.2.srf"/>
</FILES>
</DATA_BLOCK>
```

**CGHub supplied Software**

The CGHub software distribution consists of the following four programs.

1. **gtupload** program submits the Analysis Object to CGHub and creates the object in the repository.
2. **GeneTorrent** program provides reliable and protected transfer of the sequence data to CGHub.
3. **cgquery** program provides a means to query the state of submissions at CGHub.
4. **Validation Web Services** allow centers to test their metadata prior to submission.

### 6.3 gtupload

The gtupload python script accepts the user credentials along with a directory path containing the XML and BAM files. The script should run on any machine with a python interpreter and the required prerequisites (See 3.2.1 Prerequisites on page13). The user is encouraged to modify the script or use as an example.
Table 6.15: Usage: gtupload [options]

<table>
<thead>
<tr>
<th>Options</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>--version</td>
<td>show program’s version number and exit</td>
</tr>
<tr>
<td>-h</td>
<td>–help. show this help message and exit</td>
</tr>
<tr>
<td>-u</td>
<td>ANALYSIS_DIR, –uuid=ANALYSIS_DIR path name to analysis directory</td>
</tr>
<tr>
<td>-o</td>
<td>OUT_FILE, –output-xml=OUT_FILE file in which to store raw XML output.</td>
</tr>
<tr>
<td>-s</td>
<td>SERVER, –server=SERVER CGHub server location, e.g. <a href="https://cghub.ucsc.edu">https://cghub.ucsc.edu</a></td>
</tr>
<tr>
<td>-c</td>
<td>CREDENTIAL, –credential=CREDENTIAL file containing the CGHub credential; not required with the –validate-only option</td>
</tr>
<tr>
<td>-v or -vv</td>
<td>–verbose, enable verbose output</td>
</tr>
<tr>
<td>--validate-only</td>
<td>use this if only validation is to be done.</td>
</tr>
</tbody>
</table>

With the --validate-only flag, the script will submit the files to CGHub for validation but will not store them in the metadata database.

If successful, gtupload will create the manifest.xml file used by GeneTorrent to upload the bam.

If there are errors, they will be displayed along with descriptive text to help troubleshoot the problem. The optional –o flag can be used to save the error list in an XML file.

Example of a submitted BAM that has been accepted:

```bash
$gtupload -c ~/mykey.pem -u 888a173d-8438-4d36-8529-bd50719f0f48
Welcome to gtupload version 1.07, submission software
Submission results
Submission Passed: Submission Accepted
```

### 6.4 GeneTorrent

The GeneTorrent client uses a secure, high-speed parallel transfer protocol to upload all the files in the manifest file onto CGHub. GeneTorrent runs on any POSIX-compliant machine, and accepts the same authentication credential used by gtupload.

Example:

```bash
$GeneTorrent -c ~/mykey.pem -u 888a173d-8438-4d36-8529-bd50719f0f48/manifest.xml
Generating a 1024 bit RSA private key..............++++++.++++++
writing new private key to ’/tmp/GeneTorrent-511-025347-1328754577/888a173d-8438-4d36-8529-bd50719f0f48.key’
```
6.5 cgquery

The cgquery python script is provided to simplify query operation as well as to be used as an example for local customization of the submission process.

The script calls the CGHub Web services AnalysisObject or AnalysisAttributes RESTful APIs and parses the response. Users can query by the center_name, the analysis alias, or any attribute within the submission XML. (See CGHub User Guide for more detail) Once the state=live, the data has passed all necessary validation and is available for download.

Example:

```bash
% cgquery analysis_id={uuid}
==================================================================
Script Version : 1.9
CGHub Server : https://cghub.ucsc.edu
REST Resource : /cghub/metadata/analysisObject
QueryString : analysis_id=4ed09e8d-ebfe-49b6-b333-5b6069d7a397
Output File : None
Results Returned : 1
==================================================================
analysis_id : 4ed09e8d-ebfe-49b6-b333-5b6069d7a397
analysis_data_url: https://cghub.ucsc.edu/cghub/data/analysis/download/4ed09e8d-ebfe-49b6-b333-5b6069d7a397

analysis_attribute_uri:
https://cghub.ucsc.edu/cghub/metadata/analysisAttributes/4ed09e8d-ebfe-49b6-b333-5b6069d7a397

last_modified : 2011-06-22T07:00:00Z
center_name : BI
state : live
aliquot_id : f39ed42b-4da7-4412-b75d-37c859470931
Files
filename : C508.TCGA-18-5595-11A-01D-1632-08.1.bam
filesize : 22266215806
checksum : 40307654d402c56ee7fce8f53238eac6
```

6.6 Validation Web Services

Users can use CGHub Web Services to validate their metadata files prior to submission using gtupload or by calling the REST resource directly. See section 5.3 for details on using gtupload.

The REST resource can be accessed via an HTTPS POST request to

https://cghub.ucsc.edu/cghub/metadata/analysis/validate

The POST body must include the contents of the input files in three separate fields: The analysisxml field must hold the contents of analysis.xml, the runxml field must hold the contents of run.xml and the experimentxml field must hold the contents of experiment.xml.
Example curl commands:

curl -k -v
-F "analysisxml=@14f45560-3800-4f09-8e32-22efc4247728/analysis.xml"
-F "experimentxml=@14f45560-3800-4f09-8e32-22efc4247728/experiment.xml"
-F runxml=@14f45560-3800-4f09-8e32-22efc4247728/run.xml
"https://cghub.ucsc.edu/cghub/metadata/analysis/validate"

POST Response

If the metadata is correct, the POST response will be HTTP:

code 200 Metadata Valid

POST Error Responses

400 Bad Request

There are errors in the submitted XML. The response will contain an XML string with all of the errors to help troubleshoot the problem.

An example error response:

<submission_set>
  <version>TCGA 1.0</version>
  <status>ERROR</status>  <files>  <status>ERROR</status>  <details>  <analysis>
    <status>ERROR</status>  <errors>
      <error>Value of //ANALYSIS_SET/ANALYSIS/STUDY_REF[refcenter] did not match with pattern ^NHGRI$</error>
      <error>Value of //ANALYSIS_SET/ANALYSIS/STUDY_REF[refname] did not match with pattern ^\(phs000178\).*$</error>
      <error>Value of //ANALYSIS_SET/ANALYSIS/TARGETS/TARGET[refname] did not match with pattern ^\[a-zA-Z0-9\]{8}-\[a-zA-Z0-9\]{4}-\[a-zA-Z0-9\]{4}-\[a-zA-Z0-9\]{4}-\[a-zA-Z0-9\]{12}$</error>
    </errors>
  </analysis>
  <experiment>  <status>ERROR</status>  <errors>
    <error>string:37:0:ERROR:SCHEMASV:SCHEMAV_ELEMENT_CONTENT: Element 'FOO': This element is not expected. Expected is one of ( LS454, ILLUMINA, HELICOS, ABI_SOLID, COMPLETE_GENOMICS, PACBIO_SMRT, ION_TORRENT ).</error>
  </errors>
  </experiment>
  <run>  <status>OK</status>  </run>
</details>  </files>
<references>  <status>OK</status>  <details>  <analysis_run>  <status>OK</status>  </analysis_run>  <run_experiment>  <status>OK</status>  </run_experiment>  </details>
<references>
</references>
</submission_set>

404 URL Not Found
The URL was not correctly formed.

500 Internal Server Error

The server encountered an unexpected condition that prevented it from fulfilling the request.