Welcome to this Training Session with Theiagen Genomics

We will soon be getting started

Theiagen®

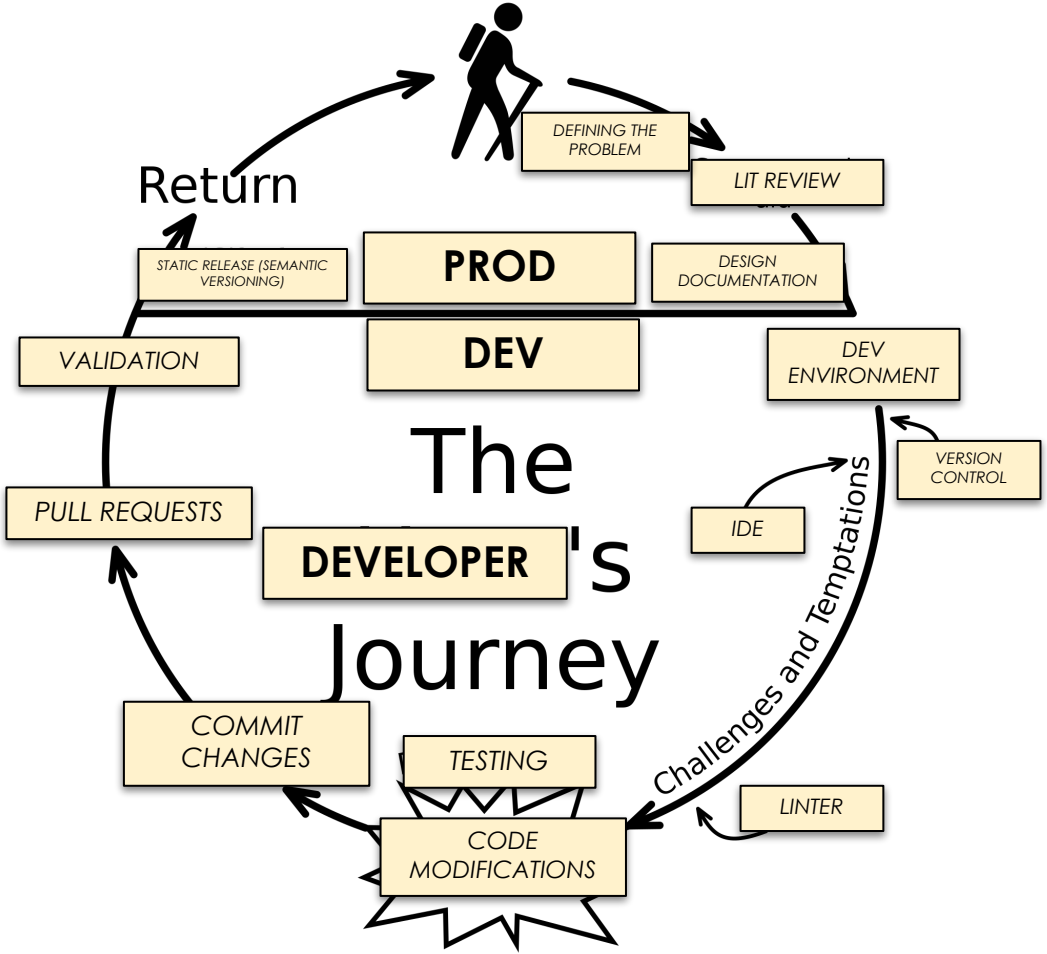
Software Development Practices for Public Health Bioinformatics

Week 04: Advanced Terra Usage

Western Region WFD Offering Provided by the
Washington State Department of Health in Collaboration with Theiagen Genomics

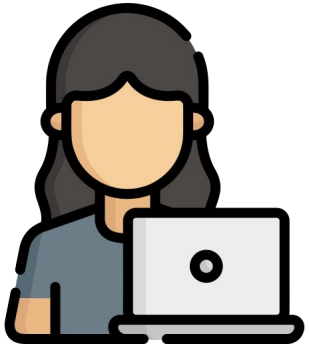
Week 1-3 Recap

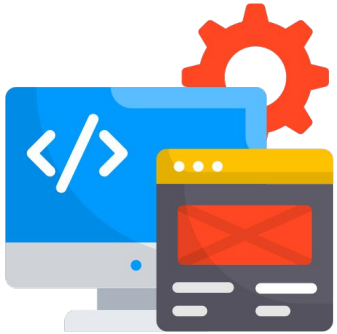
Theiagen®



The Developer's Journey

Framework where a protagonist **enters into their dev environment**, faces challenges, gains new wisdom, and **brings changes into production**.





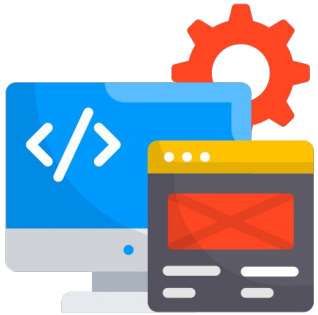
Theiagen

Byrne, C. (2017). The Hero's Journey. Wikipedia https://en.wikipedia.org/wiki/File:Heroesjourney.svg

Software Development Practices

Developer's Journey

1. **Design Document**
   a. Clearly defining the problem and the proposed solution
2. **Development Environment**
   a. Separate from production
   b. Text editors and IDE's
3. **Making Source Code Modifications**
   a. Small interactive changes (version control)
4. **Peer Review**
   a. Collaborative development teams
5. **Bringing Changes into Production**
   a. Final testing
   b. Static version releases



Theiagen

Advanced Usage of the Terra Platform

Theiagen®

Advanced Usage of the Terra Platform

Programmatic Access to the Terra Platform

- Interacting with Terra resources outside of the graphic-user interface (GUI)
  - Allows for **more flexible use of Terra functions**
  - Can be applied in WDL workflows
  - Helps connect Terra to other downstream applications such as local LIMS systems or other cloud resources

Theiagen

Terra Platform in Public Health Bioinformatics

Theiagen®

Terra in Public Health Bioinformatics

Terra.Bio

- Terra is a **cloud-based GUI bioinformatics application**
- Runs in an internet browser and enables point & click bioinformatics:
  - Upload and organize sequence data
  - Analyze data with **open-source bioinformatics workflows**
  - Access, download and share results

Terra

Theiagen®

Terra in Public Health Bioinformatics

Terra.Bio

- Has helped to **rapidly deploy bioinformatics capabilities** to public health laboratories around the world
  - As of October 2023, Theiagen workflows have been used for **5,804,839 sample analyses** by an estimated **90 PHLs representing over 40 countries**

*Facilitates **portable access**, **interoperable workflows**, and **transferable outputs** for public health scientists*

Terra

Theiagen

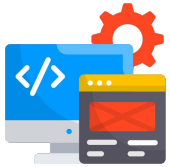
Programmatic Access to the Terra Platform

Theiagen®

Programmatic Access to the Terra Platform

Terra.Bio

- Provides a graphical user interface (GUI) that makes it easy for users to scalable **cloud resources** and **open-source workflows**
  - Currently on **GCP** and Azure is in beta
  - Uses Cromwell engine to run **WDL workflows**
    - Nextflow engine and workflow language on the Terra roadmap



Theiagen

Programmatic Access to the Terra Platform

Beyond the GUI

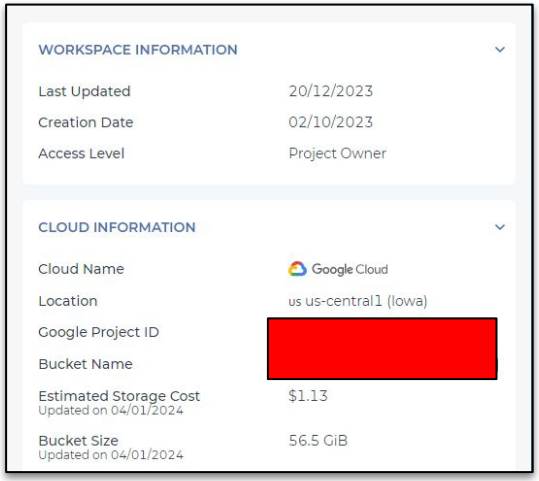
- **Direct cloud interaction**: Users can interact with Terra workspaces directly using Google Cloud Platform (GCP) tools.
- **API Access**: Terra offers a robust API (*formerly FireCloud*) for programmatic access to its features and functionalities

Theiagen

Programmatic Access to the Terra Platform

Direct Data Access through GCP

- Data within a Terra workspace is stored on a **dedicated GCP bucket**
  - This GCP bucket name is available to users on through the Workspace Dashboard



Programmatic Access to the Terra Platform

Direct Data Access through GCP

- When properly authenticated, developers can access data within this GCP bucket directly using the **gcloud command suite**
  - Helpful when importing/exporting **large data volumes**

What is the gcloud CLI?

The Google Cloud CLI is a set of tools to create and manage Google Cloud resources. You can use these tools to perform many common platform tasks from the command line or through scripts and other automation.

For example, you can use the gcloud CLI to create and manage the following:

- Compute Engine virtual machine instances and other resources
- Cloud SQL instances
- Google Kubernetes Engine clusters
- Dataproc clusters and jobs
- Cloud DNS managed zones and record sets
- Cloud Deployment Manager deployments

You can also use the gcloud CLI to deploy App Engine applications, manage authentication, customize local configuration, and perform other tasks.

Programmatic Access to the Terra Platform

Direct Data Access through GCP

- Google offers detailed documentation on how to properly install gclouds on a variety of environments (e.g. Linux, Debian/Ubuntu, Windows, etc.)
  - Once installed, use the `**gcloud auth login**` command to authenticate using the **same Google ID** used to access your workspace

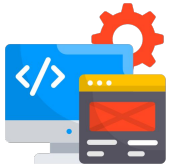
*With this setup, you will have the **full gcloud command suite** available to interact with the Terra GCP Bucket*

Theiagen

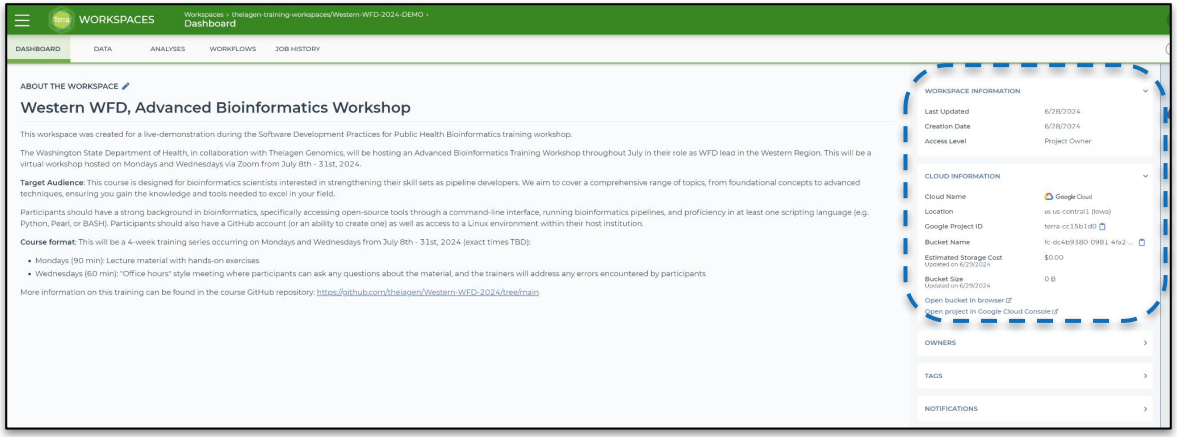
Programmatic Access to the Terra Platform

Direct Data Access through GCP

- The gcloud command suite is similar to interacting with a local directory; Common gcloud commands:
  - **gcloud storage ls {dir}** - lists items hosted in a directory
  - **gcloud storage cp {file} {dir}** - copies file(s) to a specific directory
  - **gcloud storage mv {file} {dir}** - movies file(s) to a specific directory



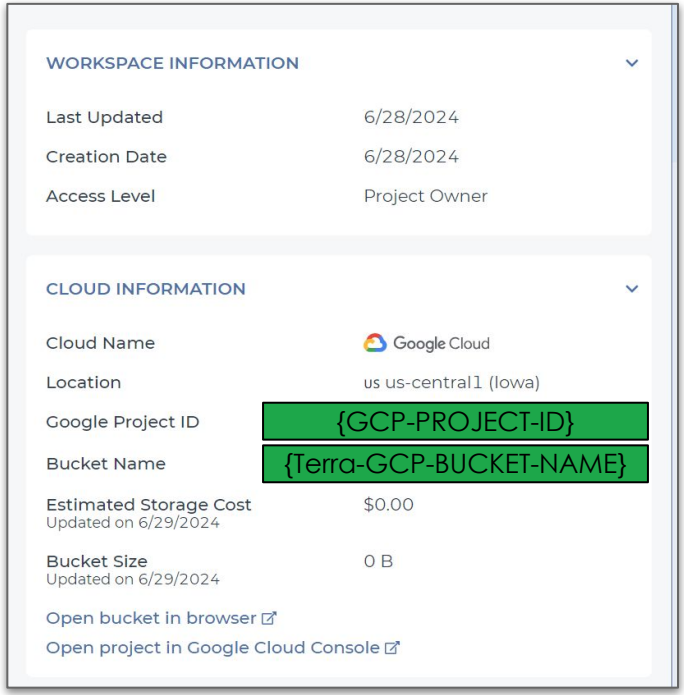
Theiagen

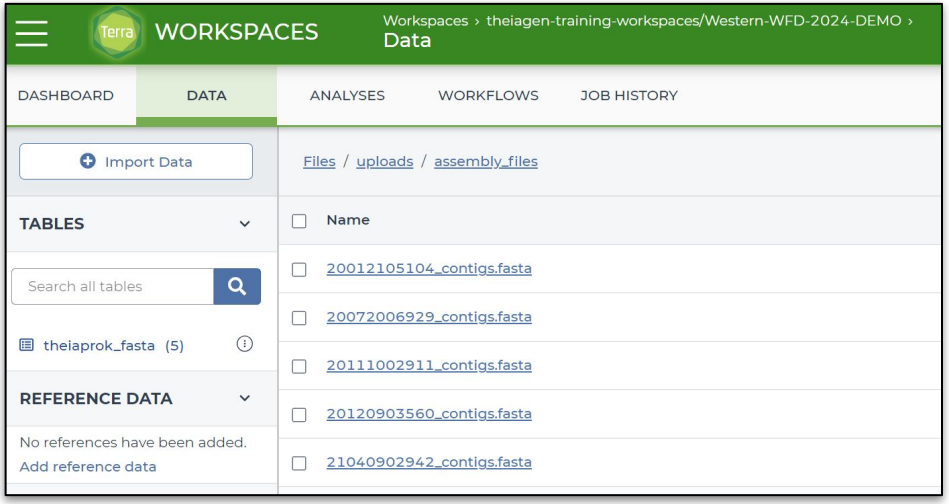


Workspace Dashboard

Cloud information: *Back-end access to the data storage*

Protected access based on Terra workspace roles; ***share cautiously***





Assembly field uploaded to Terra workspace

```
kevin_libuit@libuit-2023-dev-vm:~$ gcloud storage ls gs://{Terra-GCP-BUCKET-NAME}/uploads/assembly_files/
gs://{Terra-GCP-BUCKET-NAME}/uploads/assembly_files/20012105104_contigs.fasta
gs://{Terra-GCP-BUCKET-NAME}/uploads/assembly_files/20072006929_contigs.fasta
gs://{Terra-GCP-BUCKET-NAME}/uploads/assembly_files/20111002911_contigs.fasta
gs://{Terra-GCP-BUCKET-NAME}/uploads/assembly_files/20120903560_contigs.fasta
gs://{Terra-GCP-BUCKET-NAME}/uploads/assembly_files/21040902942_contigs.fasta
```

Programmatic Access to the Terra Platform

API Access

- Terra offers a robust API (*formerly FireCloud API*) for **programmatic access to its features and functionalities**
- API: Application Programming Interface
  - Set of rules and protocols that allows different software applications to communicate with each other

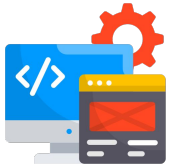
*Among many other things, APIs facilitate **automation of repetitive tasks** by enabling scripts to perform complex operations **without human intervention***

Theiagen

Programmatic Access to the Terra Platform

API Access

- Various ways to interact with Terra API
  - **Swagger UI** - interactive interface for exploring and executing API calls directly from your browser
  - **API Libraries** - Libraries and SDKs (like FISS) provide a higher-level interface for interacting with the Terra API
  - **HTTP Request** - Directly sending HTTP requests to the Terra API endpoints



Theiagen

Programmatic Access to the Terra Platform

API Access, Swagger UI

- Tool used to **visualize and interact** with the API's resources without having any of the implementation logic in place
- Presents APIs in a format that is easy to read and understand
  - Lists all available endpoints and operations, providing a clear and structured view of the API

Theiagen

Swagger
Supported by SMARTBEAR

/api-docs.yaml

Explore

Terra 0.1 OAS3

/api-docs.yaml

Terra API

Terms of service

BSD

*Helpful resource to **quickly access API calls** through a web interface*

Submissions

GET /api/submissions/queueStatus workflow queue status workflowQueueStatus

GET /api/workspaces/{workspaceNamespace}/{workspaceName}/submissions List submissions. listSubmissions

POST /api/workspaces/{workspaceNamespace}/{workspaceName}/submissions Create a submission. createSubmission

Parameters

Try it out

Name	Description
workspaceNamespace * required string (path)	Workspace Namespace workspaceNamespace
workspaceName * required string (path)	Workspace Name workspaceName

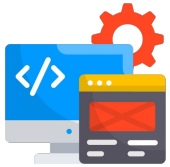
Request body required

application/json

Programmatic Access to the Terra Platform

API Access, API Libraries

- Multiple python libraries are available that provide CLI commands for for interacting with the Terra API
  - Simplifies the process by providing pre-built functions and methods for making API calls, reducing the need for manual HTTP requests



Programmatic Access to the Terra Platform

API Access, API Libraries

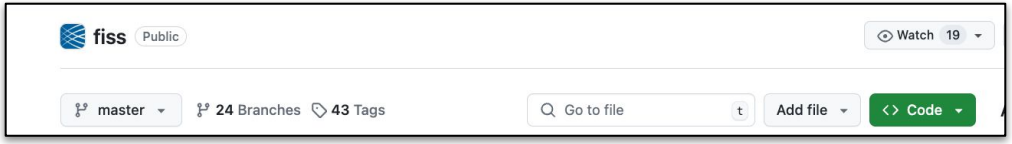
- Two commonly utilized Terra API Libraries:
  - FISS -- (Fi)reCloud (S)ervice (S)elector
    - https://github.com/broadinstitute/fiss
  - Broad terra-tools
    - https://github.com/broadinstitute/terra-tools

Theiagen

Programmatic Access to the Terra Platform

Broad's FISS -- (Fi)reCloud (S)ervice (S)elector

- Comprehensive library that provides both Python and Unix bindings to the Terra API
  - https://github.com/broadinstitute/fiss/tree/master



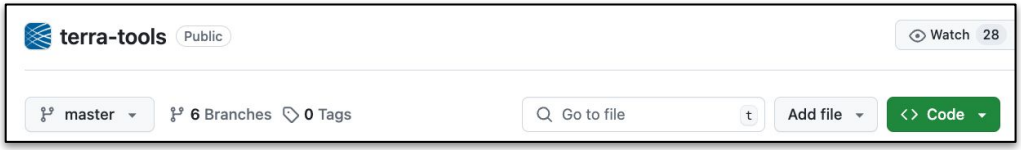

```
FISS -- (Fi)reCloud (S)ervice (S)elector

FISS is a programmatic interface to FireCloud (FC), providing a set of low- and
high-level Python bindings to the FireCloud API, as well as UNIX bindings for
command line usage.  By wrapping the FireCloud RESTful API in this manner, our
hope is to provide an interface that resonates more closely with the majority of
expected FC users--supporting interaction with FC in memes familiar to them, as
biomedical researchers & informaticians rather than database or web programmers

Like legacy FISS, the (Fi)rehose (S)ervice (S)selector that was created for
internal use at the Broad Institute, FISSFC aims to be:
```

Theiagen

Programmatic Access to the Terra Platform

Broad's Terra-Tools

- Pre-packaged scripts optimized for large data table import and export
  - https://github.com/broadinstitute/terra-tools



To run a script using Docker:

```
docker run --rm -it -v "$HOME"/.config:/.config -v "$HOME"/Documents/:/data broadinstitute/terra-tools:latest bash -c "cd data; python3 /scripts/path_to_script/<script name.py> <arguments>"
```

1. `-v "$HOME"/.config:/.config` - allows for authentication within the Docker of your Google credentials in your local $HOME directory where they are stored by default

Theiagen

Programmatic Access to the Terra Platform

Broad's Terra-Tools

Utilized heavily throughout our ***PHB repository***

repo:theiagen/public_health_bioinformatics terra-tools

Filter by

- Code 13
- Issues 1
- Pull requests 13
- Discussions 0
- Commits 6
- Packages 0
- Wikis 0

Paths

- tasks/
- tasks/utilities/
- tasks/utilities/data_export/
- tasks/utilities/data_handling/
- tasks/utilities/submission/
- More directories...

Advanced

- Owner
- Symbol
- Exclude archived

Advanced search

13 files (83 ms) in theiagen/public_health_bioinformatics

tasks/utilities/data_export/task_export_two_tsvs.wdl

```
11    Int disk_size = 10
12    Int memory = 1
13    Int cpu = 1
14    String docker = "us-docker.pkg.dev/general-theiagen/theiagen/terra-tools:2023-03-16"
15  }
16  meta {
17    # added so that call caching is always turned off
```

tasks/utilities/data_export/task_download_terra_table.wdl

```
17    String docker = "quay.io/theiagen/terra-tools:2023-06-21"
```

tasks/gene_typing/variant_detection/task_snippy_gene_query.wdl

```
6     File snippy_variants_results
7     String? query_gene
8     File? reference
9     String docker = "us-docker.pkg.dev/general-theiagen/theiagen/terra-tools:2023-06-21"
10    Int cpu = 8
11    Int memory = 32
12    Int disk_size = 100
```

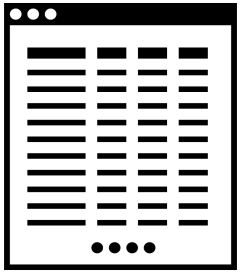
tasks/taxon_id/contamination/task_kraken2.wdl

```
168   String? target_organism
169   Int cpu = 4
170   Int disk_size = 100
171   String docker = "us-docker.pkg.dev/general-theiagen/theiagen/terra-tools:2023-08-28-v4"
172   Int memory = 8
```

Mercury Workflows

Prepare SC2 Sample Data for Submission

Read Data



Sample Metadata

```
AAAGAAACTATAGCTGAGAGCG
GCGATCGTACGATGCATGCTAG
CTAGCGAGAGCGGCGATCGTAC
GATGCATGCTAGCTAGCGAGAG
CGGCGATCGTACGATGCATGCT
AGCTAGCGAGAGCGGTACGATG
```

Genome Assembly

GISAID/NCBI Submission Files

GISAID

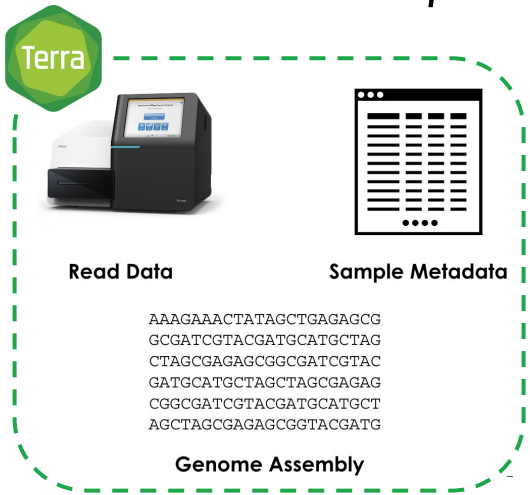
NCBI

More information on the Mercury Workflow available through our **PHB documentation**

Theiagen®

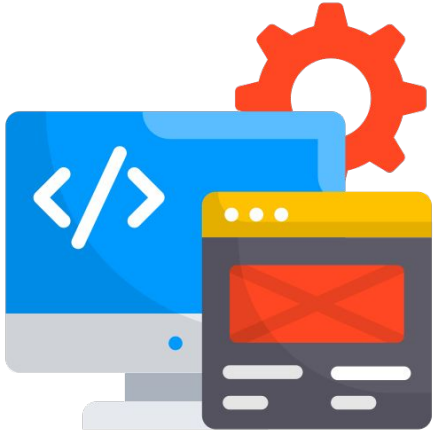
Mercury Workflows

Prepare SC2 Sample Data for Submission



*All stored in a **data table within a user's Terra workspace***

*At launch, the **data table is exported** to the VM running the Mercury workflow job using **Broad's Terra-Tools functions***



*Within the VM, we **parse the datatable** and format it according to the submission endpoint, i.e. GISAID, NCBI*

Theiagen®

Mercury Workflows

Prepare SC2 Sample Data for Submission

```
command <<<
  # when running on terra, comment out all input_table mentions
  python3 /scripts/export_large_tsv/export_large_tsv.py --project "~{project_name}" --workspace "~{workspace_name}" --entity_type ~{table_name} --tsv_filename ~{table_name}-data.tsv

  # when running locally, use the input_table in place of downloading from Terra
  #cp -v ~{input_table} ~{table_name}-data.tsv

  # transform boolean skip_county into string for python comparison
  if ~{skip_county}; then
    export skip_county="true"
  else
    export skip_county="false"
  fi
```

https://github.com/theiagen/public_health_bioinformatics/blob/f9b8070/tasks/utilities/submission/task_mercury_file_wrangling.wdl

Example of a WDL workflow ***utilizing the Terra API*** *for a specific use case*

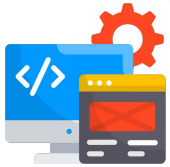
Theiagen®

Programmatic Access to the Terra Platform

API Access, HTTP Requests

- Provides the most control and flexibility, suitable for custom integrations and detailed interactions with the API
  - Requires knowledge of HTTP methods (GET, POST, PUT, DELETE) and handling of request headers and payloads
- Tools like curl can be used to transfer data with URLs

*Highest technical difficulty and **most customizable approach** to interacting with the Terra API*



Theiagen

Programmatic Access to the Terra Platform

Terra API HTTP Request Example

```
# submit job
curl -X 'POST' \
  "https://api.firecloud.org/api/workspaces/${DESTINATION_PROJECT}/${DESTINATION_WORKSPACE}/submissions" \
  -H 'accept: */*' \
  -H "Authorization: Bearer ${TOKEN}" \
  -H 'Content-Type: application/json' \
  -d "{
  \"methodConfigurationNamespace\": \"${DESTINATION_PROJECT}\",
  \"methodConfigurationName\": \"TheiaProk_Illumina_PE_PHB\",
  \"entityType\": \"${DESTINATION_TABLE}_set\",
  \"entityName\": \"${tableName}-${TODAY_DATE}\",
  \"expression\": \"this.${DESTINATION_TABLE}s\",
  \"useCallCache\": false,
  \"deleteIntermediateOutputFiles\": false,
  \"useReferenceDisks\": false,
  \"memoryRetryMultiplier\": 1,
  \"workflowFailureMode\": \"NoNewCalls\",
  \"ignoreEmptyOutputs\": true,
  \"userComment\": \"${tableName}-${TODAY_DATE} job automatically launched\"
  }"
```

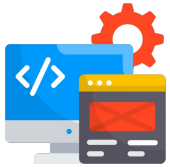
Example HTTP Request for launching a Terra workflow

Theiagen

Programmatic Access to the Terra Platform

Summary

- Developers can interact with Terra workspaces directly using **Google Cloud Platform (GCP) tools**, e.g. gcloud
  - Helpful for large data import/export
- **Terra API** offers programmatic access to its features and functionalities
  - Can be accessed through **Swagger UI**, **API libraries**, or **direct HTTP requests**



Theiagen

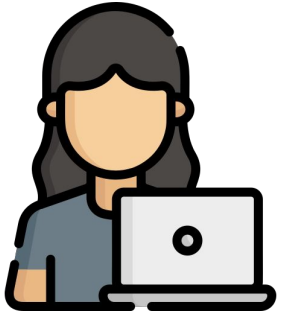
Live Demo

Theiagen®

Non-GUI Terra Usage

Demo Goal

1. **Uploading data to a Terra Workspace**
   a. Using gcloud to transfer data directly to a Terra-accessible GCP Bucket
2. **Utilizing Terra API**
   a. Swagger UI call
   b. Terra-Tools import/export table demo
   c. HTTP Request to launch a workflow



Theiagen