



Jul 28, 2021

# Auger/Auspice/UShER: SARS-CoV-2 Cluster Detection Workflow for the Terra Platform

Forked from [Titan Auger SARS-CoV-2 Cluster Detection Workflow for the Terra Platform](#)

Anusha Ginni<sup>1</sup>, Jill V Hagey<sup>1</sup>

<sup>1</sup>Centers for Disease Control and Prevention

1 Works for me

 Share

[dx.doi.org/10.17504/protocols.io.bv4kn8uw](https://dx.doi.org/10.17504/protocols.io.bv4kn8uw)

Anusha Ginni

## DISCLAIMER

The opinions expressed here do not necessarily reflect the opinions of the Centers for Disease Control and Prevention or the institutions with which the authors are affiliated. The protocol content here is under development and is for informational purposes only and does not constitute legal, medical, clinical, or safety advice, or otherwise; content added to [protocols.io](#) is not peer reviewed and may not have undergone a formal approval of any kind. Information presented in this protocol should not substitute for independent professional judgment, advice, diagnosis, or treatment. Any action you take or refrain from taking using or relying upon the information presented here is strictly at your own risk. You agree that neither the Company nor any of the authors, contributors, administrators, or anyone else associated with [protocols.io](#), can be held responsible for your use of the information contained in or linked to this protocol or any of our Sites/Apps and Services.

## ABSTRACT

The Titan\_Augur workflows was written to process concatenated assembly fasta data for SARS-CoV-2 Phylogenetics analysis and cluster detection.

Augur has two workflows, Augur\_Prep and Augur\_Run which you need to run one after the other. The output of these workflows is the required inputs for Auspice/Nextstrain to understand the genetic relatedness and draw phylogenetics inferences.

The inputs for the Augur\_Prep job include metadata on the sample (you can upload as a tsv), the assembly fasta and pango lineage for each sample. If you ran Titan on Terra you will already have generate the the assembly fasta and pango lineage for samples, but if you used another pipeline you will need to upload them.

Upon initiating a Augur\_Prep job, input data will be processed and the output will be a sample metadata and assemblies in the correct format to perform Augur\_run analysis. The Augur\_run workflow then takes the metadata and the concatenated fasta assemblies to generate the outputs that can be used for phylogenetics inferences on Auspice or Nextstrain visualization interface tools.

This protocol also shows steps on how to visualize using the Auspice as well as UCSC UShER tools.

Here are some helpful viedos:

- [Theiagen training video on Augur](#)

<https://www.youtube.com/watch?v=r92L-B6wWOU>

<https://www.youtube.com/watch?v=4nPS5Vab2dE>

<https://www.youtube.com/watch?v=uDXBcHR8GSQ&t=2186s>

DOI

[dx.doi.org/10.17504/protocols.io.bv4kn8uw](https://dx.doi.org/10.17504/protocols.io.bv4kn8uw)

PROTOCOL CITATION

Anusha Ginni, Jill V Hagey 2021. Auger/Auspice/UShER: SARS-CoV-2 Cluster Detection Workflow for the Terra Platform. [protocols.io](#)  
<https://dx.doi.org/10.17504/protocols.io.bv4kn8uw>



FORK NOTE

FORK FROM

[Forked from Titan Auger SARS-CoV-2 Cluster Detection Workflow for the Terra Platform, Frank Ambrosio](#)

KEYWORDS

Nanopore, SARS-CoV-2, Pangolin, Genomics, Virology, RNA, Covid, Computational Biology, Sequencing, Phylogenetics, Auspice, NextStrain, GISAID, UShER, Augur

LICENSE

This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

CREATED

Jun 25, 2021

LAST MODIFIED

Jul 28, 2021

PROTOCOL INTEGER ID

51052

DISCLAIMER:

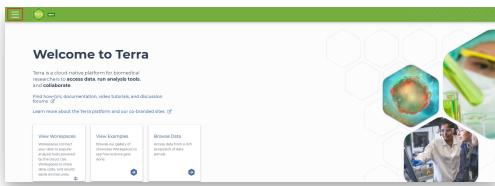
The opinions expressed here do not necessarily reflect the opinions of the Centers for Disease Control and Prevention or the institutions with which the authors are affiliated. The protocol content here is under development and is for informational purposes only and does not constitute legal, medical, clinical, or safety advice, or otherwise; content added to [protocols.io](#) is not peer reviewed and may not have undergone a formal approval of any kind. Information presented in this protocol should not substitute for independent professional judgment, advice, diagnosis, or treatment. Any action you take or refrain from taking using or relying upon the information presented here is strictly at your own risk. You agree that neither the Company nor any of the authors, contributors, administrators, or anyone else associated with [protocols.io](#), can be held responsible for your use of the information contained in or linked to this protocol or any of our Sites/Apps and Services.

Setup Terra and Google Cloud Accounts

1

The Terra platform registration requires a Google account. If you have a Google account you can sign in using the Terra login page:

<https://app.terra.bio/>



Welcome page for Terra.bio.

If you do not have Google email you can set up a Google account with a non-Google email. The steps to do this are described in the following link:

<https://support.terra.bio/hc/en-us/articles/360029186611>

The Terra platform uses the Google Cloud to run workflows and store data. The following documentation will describe how to set up a Google Cloud account:

<https://support.terra.bio/hc/en-us/articles/360046295092>

To link your Terra platform account with your Google Cloud account follow the instructions provided in the following link:

<https://support.terra.bio/hc/en-us/articles/360026182251-How-to-set-up-billing-in-Terra>

Scroll down to Section 3 titled "Create a Terra Billing Project" and follow the instructions. **It is important to note that the name you give to your Terra Billing Project must be unique across all Google Billing Projects.** If the name provided is not unique, it won't immediately throw an error, but instead will not complete the process of associating the Google Billing account with the Terra Billing Project. If this occurs cancel the process and set up a new Terra Billing Project.

For detailed step by step instructions, refer to Terra registration check out our [protocol on registration and billing](#).

## Import Augur workflows from Dockstore

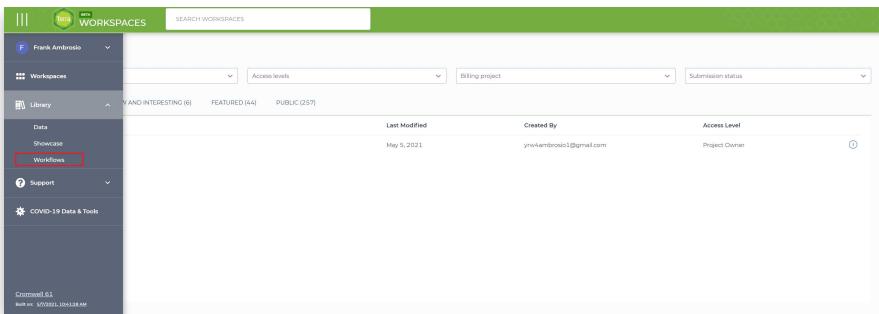
2

### Importing the Titan\_Augur Workflows from Dockstore to the Terra User Workspace

**Note:** Augur workflow includes 2 pipelines: 1-Augur\_Prep, 2-Augur\_Run.

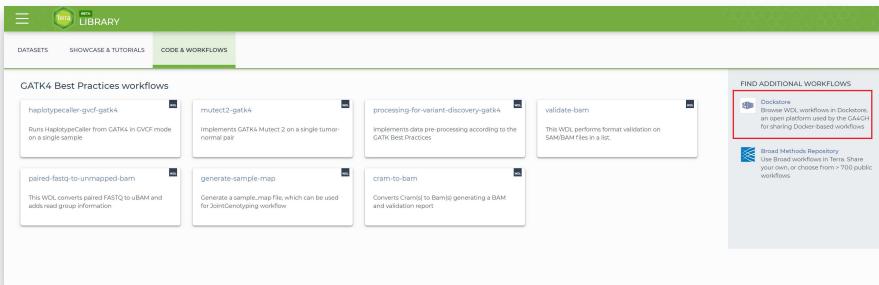
The Augur\_Prep workflow prepares the metadata that is required to run the Augur\_Run workflow, so it is required to import both the pipelines into your workspace.

The Augur workflows is hosted on the [Dockstore repository](#) and has to be imported into the user's Terra Workspace. Begin by clicking on the three parallel lines in the top left-hand corner, followed by clicking the 'Library' tab and finally click the 'Workflows' button.



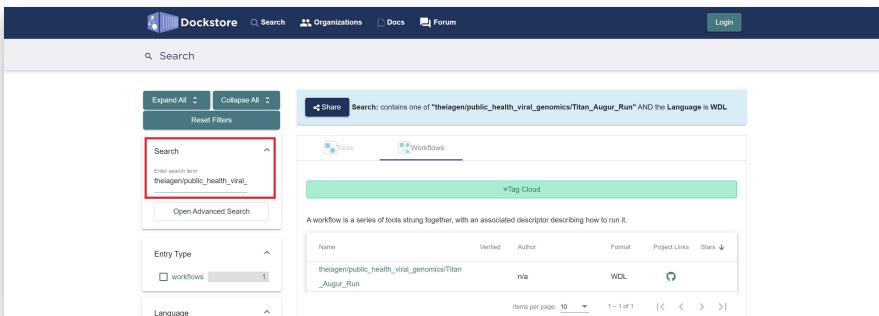
'Workflows' button listed under the 'Library' tab in the selection panel

On the right side of the page Under 'Find Additional Workflows' select the 'Dockstore' from the dialogue box .



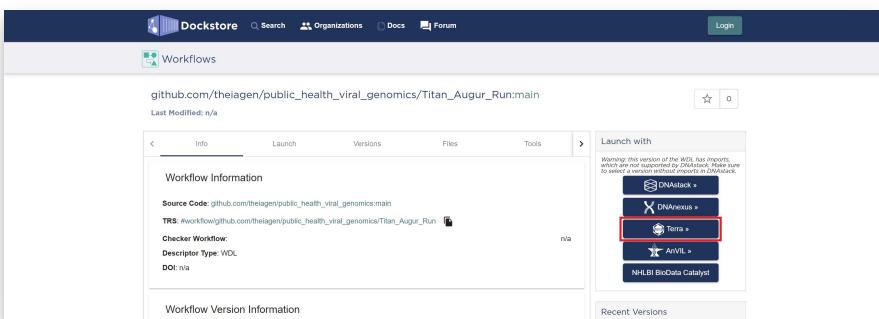
Workflows panel with link to Dockstore

On the left side of the Dockstore page search for 'theiagen/public\_health\_viral\_genomics/Titan\_Augur\_Run' in the search bar.



Search results for theiagen/public\_health\_viral\_genomics/Titan\_Augur\_Run

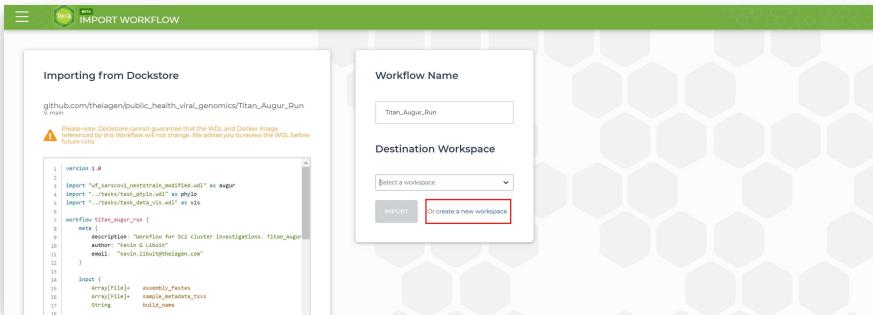
Click the 'theiagen/public\_health\_viral\_genomics/Titan\_Augur\_Run' link. This will take you to a page where you can import the workflow into your Terra workspace.



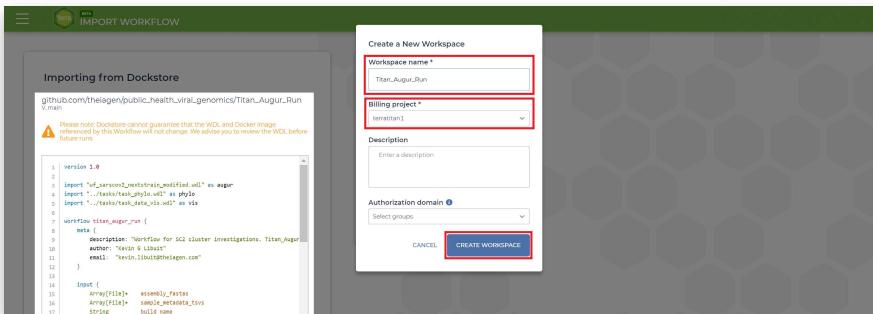
Dockstore link to terra

On the right-hand side of the page under the 'Launch with' window click the 'Terra' button. It should bring you back to the Terra platform within the 'Import Workflow' page. If you already have a workspace select it from the dropdown menu Under 'Destination Workspace' and click 'Import'. If not you will need to create a workspace first (see note below)

If you need to create a new workspace click the 'create a new workspace' button.

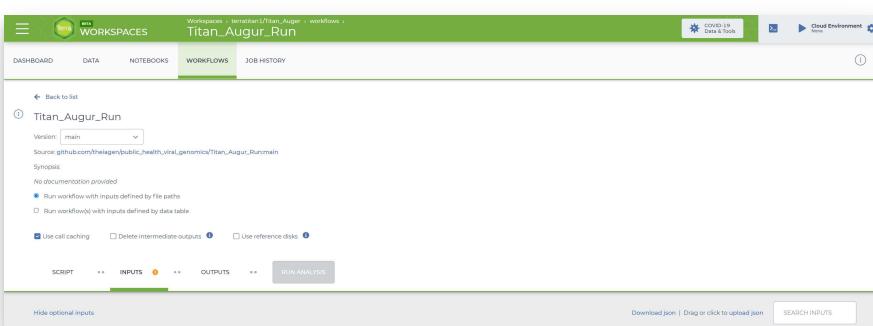


A pop-up window titled 'Create a New Workspace' should appear. Name your new workspace and associate it with a billing account using the 'Billing project' drop-down menu (this is the Terra Billing Account you created in the previous step should be available). Finally, click the 'Create Workspace' button.



#### New Workspace Panel

After clicking the 'Create Workspace' button you should be automatically directed to the Augur workflow panel in the new workspace page that was just created.



If you have imported the workflow, you should see this page.

After clicking the 'Import' button you should be automatically directed to the Augur workflow panel in your specified workspace.

If you have imported the workflow, you should see this page.

Now that you have imported the Augur\_Run workflow, follow the same steps to import the [theiagen/public\\_health\\_viral\\_genomics/Titan\\_Augur\\_Prep](#) workflow as well.

Quick video showing how to import Augur workflows (Both Augur\_Prep & Augur\_Run).

### Analysis Inputs/Data uploading

3

**Before you start: If your dataset is around <15 samples some level of genetic diversity and collection-date diversity is required to ensure that the Augur workflow runs to completion. You can achieve this by adding an outlier sample with a discrepant lineage and collection date. Its a good idea to always include a divergent lineage to ensure you have genetic and collection-date diversity.**

The Augur workflows require the assembled fasta sequence (consensus sequence), pango linages and metadata for each sample. If we look at the inputs of the workflow we will see the information we need.

Task name	Variable	Type	Attribute
titan_augur_prep	assembly	String	Required
titan_augur_prep	collection_date	String	Required
titan_augur_prep	iso_continent	String	Required
titan_augur_prep	iso_country	String	Required
titan_augur_prep	iso_state	String	Required
titan_augur_prep	pango_lineage	String	Required

Augur\_Prep workflow page where you will import all the inputs from existing workspace

The assembly file is generated from your genomic characterization workflow. If you have run Titan workflows on Terra then you will already have generated the assembly fasta files and all you need at this point is to select the samples from your analysis sample sets within your workspace. We will do this in step 4, but will still need to upload your metadata for each sample if that is not already associated with them.

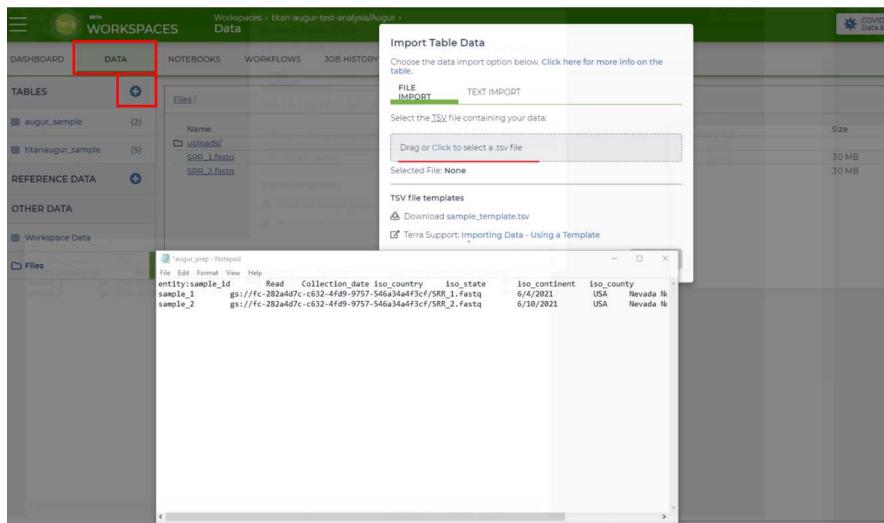
Construct a file with the following columns that are [required metadata](#) for each sample of interest:

- Collection\_date

- iso\_country
- iso\_state
- iso\_continent
- iso\_county

Here is an example for TABLE file that you will upload to your workspace: [augurprep\\_metadata.tsv](#)

To associate your metadata with the assembly fasta and pango lineages already in Terra, being by going to the DATA page and click the blue "+" sign next to the TABLES tab on the left of the web page. A pop-up window will appear to import a table file (see example file above) and either drag and drop the file or browse to find it. Once uploaded return to your samples and notice you will now how additional columns associated with each sample. **Terra will know to associate the columns with the correct row by your "entity:" column so use the same sample\_ids as you did when you originally uploaded the data.**



Uploading metadata to associate with samples already run through Titan on Terra.

**If you have already performed some genome characterization and all the metadata is associated with your samples, follow step 4 to select the Augur\_Prep & Augur\_Run required inputs from the TABLE (detailed instructions in step 4).**

If you did not perform any genomic characterization of SARS-CoV-2 sequences before on Terra, you will need to upload your assemblies, and its associated metadata file to run the Augur workflows (see sub-steps below).

You have 3 options for uploading your assemblies and metadata. These are the same as what you saw in the Titan protocols.

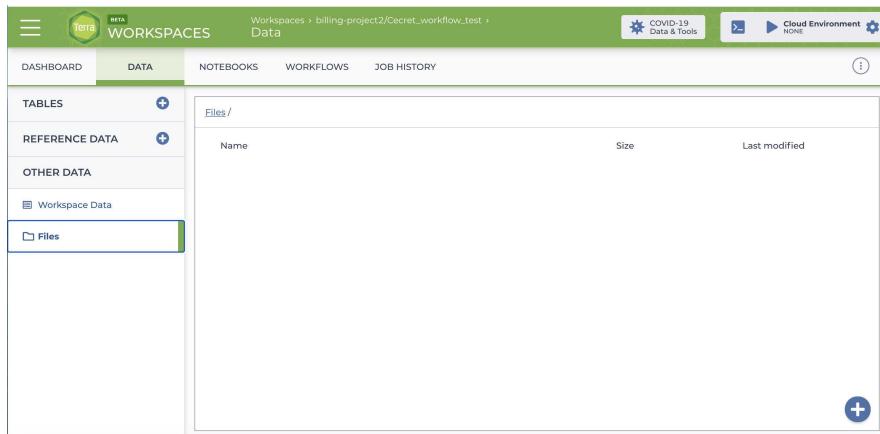
1. Upload on Terra (can only do single files).
2. Upload to Google bucket and link to Terra (can do single files or bulk files/folders).
3. Upload via '<https://app.terra.bio/#upload>' – **This is the easiest option**

Here we will show you how to select/upload metadata information to associate with your samples.

**Note: All the variable names and metadata information in the instructions are dummy and are just for demo purpose only. Please make sure you are using the right variables/names that pertain to your samples.**

### 3.1 The first option is to upload a single sample at a time:

Inside your new workspace page, click on the 'Data' panel in the newly created workspace and then click on the 'Files' tab.



The 'Files' tab within the 'Data' panel of the newly created workspace

Once in the 'Files' tab you can either just drag and drop your files into this space or move the mouse over the blue plus sign icon in the bottom right-hand corner and click 'upload'. Upload the sequence files you'll need for this analysis. **Each fastq sequence file has to be uploaded individually using this method.**

Once the files are uploaded you will need to bring a table in to associate your files with their corresponding link to their google bucket location.

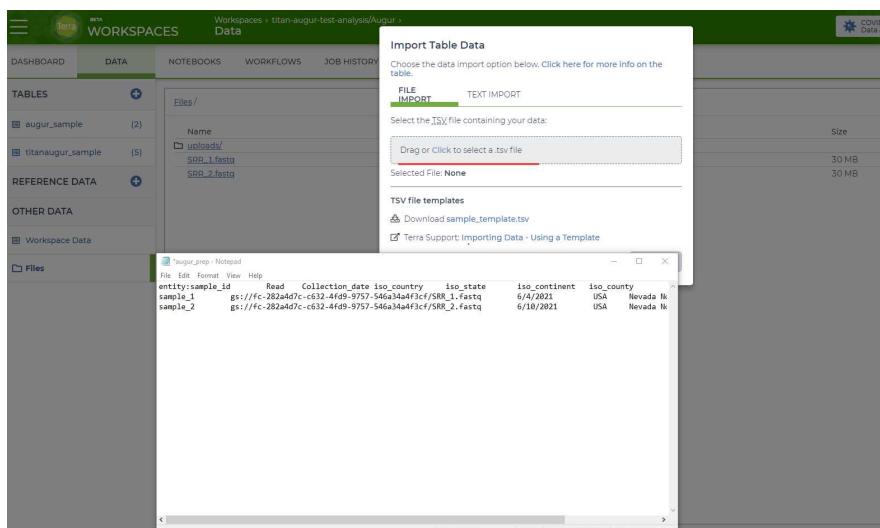


TABLE data updated with the file paths before uploading to workspace.

The Terra sample table file has to follow a specific template. We've provided the template file here [Terra\\_augur\\_Table\\_Upload.txt](#) as a downloadable **tab-separated** (or .tsv) file. This example file has the **minimum amount** of columns to be able to create a collection using this method.

The tab-separated table has two columns since we have ONT data in this example:entity:sample\_id and fastas. If you had Illumina PE data you would need three columns entity:sample\_id, Forward\_Read, and Reverse\_Read.

Either by editing the text file or using spreadsheet software like Excel, fill in each column with the

required information. The first column 'entity:Test\_augur\_sample\_id' is the sample name that is provided by the user. The second column, fasta are the file paths where it is stored within the Google Cloud.

**While you had to upload your samples individually you can have all your samples in one datatable.**

To identify the Google Cloud location right-click on each fastq file that was uploaded in the previous step and copy the link address. The file path should look like something similar to the following:

gs://fc-b1e3191a-3d9f-43fe-9743-255551ce2f38/SRR11953697.fastq.gz

Once the table is filled in with the required information be sure to save it as a **tab-separated** (or tsv) file. The spreadsheet software should have an option to save as a 'tsv' file.

**Important note on column names: DO NOT USE SPACES! As we did before in creating our workspace and collection names use "\_" instead of spaces! The first column MUST have the name "entity:sample\_id", you can add other things between "entity:" and "sample\_id" if you want. Example: "entity:Illumina\_sample\_id". You can call the other columns whatever you like, but obviously clarity is key.**

When completed the table should look similar to the following:

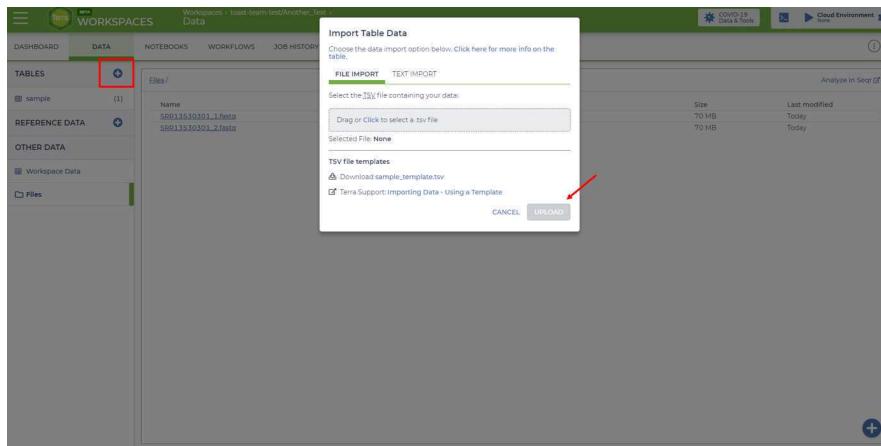
A	B	C
1 entity:Test_augur_sample_id	fastas	
2 ERR6000261	gs://fc-7312f3f9-5686-4ec3-bcee-51dd2fb5dea2/uploads/augur_data/ERR6000261.fasta	
3 ERR6000262	gs://fc-7312f3f9-5686-4ec3-bcee-51dd2fb5dea2/uploads/augur_data/ERR6000262.fasta	
4		
5		
6		
7		
8		

Example Terra sample file

Once the table is filled in with the required information be sure to save it as a **tab-separated** (or tsv) file. The spreadsheet software should have an option to save as a 'tsv' file.

**Note: When uploading additional sample table files to the same workspace the entity types must be unique and end in "\_id" (e.g. sample1\_id, sample2\_id etc.)**

Finally, the completed Terra sample file will need to be uploaded to the workspace. On the right-hand side of the workspace 'Data' panel there is a 'Tables' tab. Click the blue plus sign icon on the right edge of the 'Tables' tab. A popup window should appear titled 'Import Table Data'. Select your completed tab-separated sample file for upload and then click the 'Upload' button.



The 'Import Table Data' window for uploading the Terra sample file

If the upload is successful then the sample file should be located under the 'Tables' tab as 'sample (#)' where # is the number of samples in your file.

		CollectionDate	isoContinent	isoCountry	isoCounty	isoState
	augur_sample	6/4/2021	North America	USA	Carrizo City	Nevada
	titaneugur_sample	6/10/2021	North America	USA	Clark County	Nevada

The workspace 'Data' panel after successfully uploading the fastq sequence files and TABLE file

The whole process is shown in the video below:

You will notice in the video that you need to upload a file with sample metadata and the path to the fasta/fastq files. Once you have uploaded the files into the workspace, you can either copy, paste and save the paths to the metadata/Table file and upload or edit the paths once you have uploaded the metadata (this way is shown in the video)

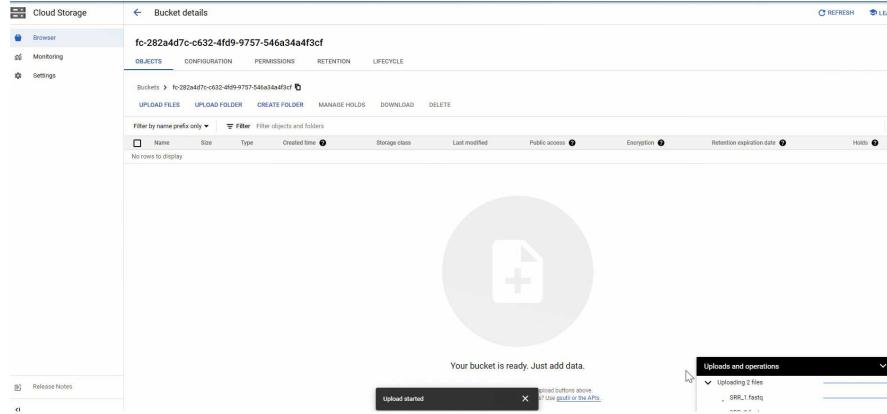
- 3.2** You will likely have many samples to upload and you can do this by going directly to your Google bucket.

First, go to "DASHBOARD" tab in your workspace and click "Google Bucket" at the bottom right corner of the same page.

The Dashboard tab of your workspace.

This will direct you to your “Google Cloud Platform” page for data uploading.

Click “UPLOAD FILES” in the middle of this page to upload single or multiple fastq files. Or you can click “UPLOAD FOLDER” to upload a folder with multiple fastq files stored inside.



Go back to your Terra account and click “DATA” tab. The successfully uploaded fastq files will show up. If the upload is successful then the files should be located under the 'Files' tab, either inside the folder (as you named) or files

Name	Size	Last modified
SR0_1.fastq	30 MB	Yesterday
SR0_2.fastq	30 MB	Yesterday

Files tab under DATA section showing uploaded files

Now that the samples are uploaded you will need to return and follow the steps in 3.0 to get your metadata associated with each sample and that should look like below.

	Collection_date	iso_continent	iso_country	iso_county	iso_state
sample_1	6/4/2021	North America	USA	Carson City	Nevada
sample_2	6/10/2021	North America	USA	Clark County	Nevada

TABLE data showing metadata of the samples.

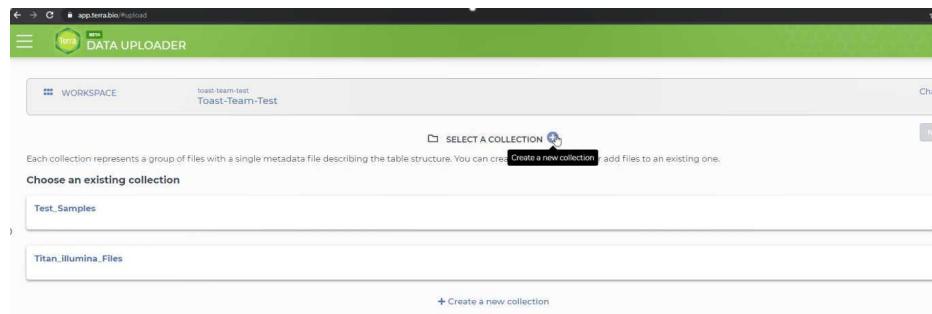
Here is a short video demonstration:

- 3.3** Alternatively, you can upload your files via '<https://app.terra.bio/#upload>'. There is no button on Terra to take you to this page, you will need to type this into the search bar.

Navigate to '<https://app.terra.bio/#upload>'

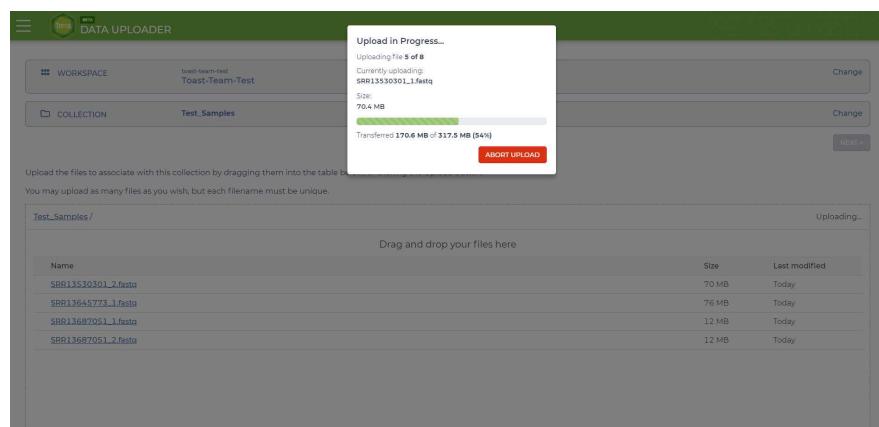
Select the Terra Workspace to which you would like to upload your fastq files. This will be the same workspace created in the previous step.

Click the '+ Create a new collection' link and enter a name for your new collection of fastq files. **DO NOT INCLUDE SPACES IN THE COLLECTION NAME, use underscores instead. Spaces will cause an error later in the pipeline.**



Click the 'Create Collection' button and you will be routed to the data uploader page where you can drag and drop fastq files directly into your browser window to initiate upload.

Drag and drop the fastq files that you would like to upload in to the upload space.



Screen when files are uploading

Once your files have been successfully uploaded, select "NEXT>" to proceed to the metadata upload page.

DATA UPLOADER

WORKSPACE titan-augur-test-analysis Augur Change

COLLECTION augurdata Change

DATA FILES Includes 5 files Change

UPLOAD YOUR METADATA FILES

Upload a tab-separated file describing your table structures.

- Any columns which reference files should include just the filenames, which will be matched up to the data files in this collection.
- The first column must contain the unique identifiers for each row. The name of the first column must start with `entity:` followed by the table name, followed by `_id`.

For example, if the first column is named `entity:sample_id`, a table named "sample" will be created with "sample\_id" as its first column. There are no restrictions on other columns.

Drag and drop your metadata .tsv or .txt file here +

Drag and drop your table data

DATA UPLOADER

WORKSPACE titan-augur-test-analysis Augur Change

COLLECTION augurdata Change

DATA FILES Includes 5 files Change

UPLOAD YOUR METADATA FILES

Creating a new Table: titan\_augur\_sample

If this table looks right to you, click the button on the right to create the table in your workspace.

entity:titan_augur_sample_id	Read	Collection_date	Iso_country	Iso_state	Iso_continent
sample_1	SRR_1.fasta	6/4/2021	USA	Nevada	North America
sample_2	SRR_2.fasta	6/10/2021	USA	Nevada	North America
sample_3	SRR_3.fasta	6/10/2021	USA	NorthCarolina	North America
sample_4	SRR_4.fasta	6/12/2021	USA	Nevada	North America
sample_5	SRR_5.fasta	6/12/2021	USA	Ohio North America	Carson City

CANCEL CREATE TABLE

DATA UPLOADER page after creating new table metadata for the samples

Here is a video of the process:

---

## Running the Augur Workflows

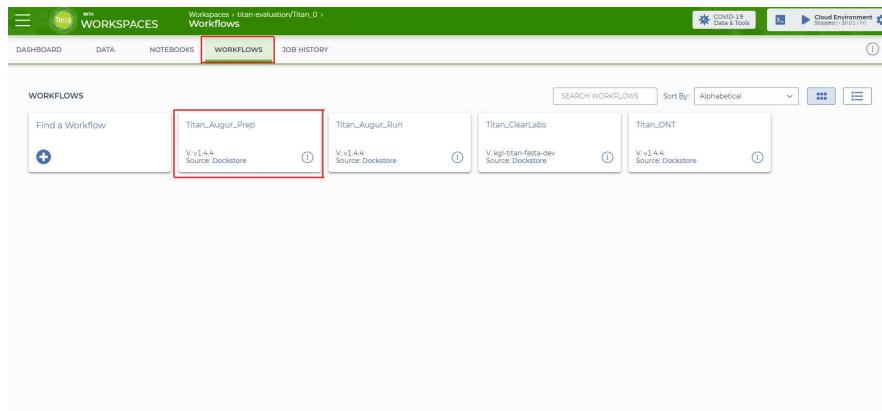
- 4 Running Titan\_Augur workflows is a two step process.
1. Run Augur\_Prep workflow
  2. Run Augur\_Run workflow

### 4.1

### Running Augur\_Prep workflow

Navigate to the Augur Prep workflow page:

From 'DASHBOARD' menu -> select 'WORKFLOWS' -> select 'Titan\_Augur\_Prep' workflow tile



This will bring up the Titan\_Augur\_Prep workflow page:

Task name	Variable	Type	Attribute
titan_augur_prep	assembly	String	Required
titan_augur_prep	collection_date	String	Required
titan_augur_prep	iso_continent	String	Required
titan_augur_prep	iso_country	String	Required
titan_augur_prep	iso_state	String	Required
titan_augur_prep	pango_lineage	String	Required

Augur\_Prep workflow page where you will import all the inputs from existing workspace

Select the version of the workflow you would like to run. If no preference pick the latest stable version (in this example v1.4.4). **The main and dev versions of the pipeline are under development and actively changing so its recommended NOT to use these versions.**

Ensure that "Run workflow(s) with inputs defined by data table" is selected, in the Output tab select 'use\_defaults', click the "Use call caching" box and then select the root entity type for the sample data you wish to analyze. **Don't use the "set" version!**

Call caching allows Terra to identify and skip jobs that have been run previously; this option is by default enabled to avoid unnecessary compute costs. More information on Terra call caching, including examples of when you may want to disable this feature, is available through the [Terra Support Documentation](#).

Click "SELECT DATA" and choose the samples you wish to analyze. **Note that where are only 25 samples shown by default so make sure when you click the check box to select all (as shown in video) that you actually get all the samples you intend to!**

For the inputs to the workflow the top six rows in the input represent variables that have to be provided by the user. Complete the INPUTS section with the appropriate attributes. In our example, for the 'assembly\_fastas' variable, for the 'Attribute' text box we set that to 'this.assembly.fasta' to indicate the 'fasta\_files' we wish to analyze. The rest of the inputs are as follows:

- 'collection date' should be 'this.collection\_date'
- 'iso\_continent' should be 'this.iso\_continent'

- 'iso\_country' should be 'this.iso\_country'
- 'iso\_state' should be 'this.iso\_state'
- 'pango\_lineage' should be 'this.pango\_lineage' --> like assembly fastas this obtained from the previous analysis (Titan or uploaded from your own).

Once you input all the data, it would look similar to the below picture.

Task name	Variable	Type	Attribute
titan_augur_prep	assembly	String	this.assembly_fastas
titan_augur_prep	collection_date	String	this.collection_date
titan_augur_prep	iso_continent	String	this.iso_continent
titan_augur_prep	iso_country	String	this.iso_country
titan_augur_prep	iso_state	String	this.iso_state
titan_augur_prep	pango_lineage	String	this.pango_lineage
prep_augur_metadata	CPUs	Int	Optional

Once you are at this page and ready, hit Run and Launch the Analysis

**NOTE:** If you named your columns something other than fastas then just type "this." followed by whatever the column name is. We would advise naming your fastas column "fastas" for clarity.

Video showing on how to submit the Augur\_Prep Run:

Once the Augur\_Prep workflow runs successfully, you should able to see the sample\_metadata.tsv file associated to the sample table data under the TABLE section. This sample\_metadata.tsv is selected as one of the inputs to run the Augur\_Run workflow, that is why this step is necessary to check it was generated first.

The video below shows this step:

## 4.2

### Running Augur\_Run workflow

As we did before select the version of the workflow you would like to run (pick the latest stable version, here v1.4.4). **The main and dev versions of the pipeline are under development and actively changing so its recommended NOT to use these versions.**

Ensure that "Run workflow(s) with inputs defined by data table" is selected and click the "Use call caching" box. **In contrast to the Augur\_prep workflow, select "augur\_prep\_sample\_set" in the the root entity section, because it will expect to have set of files** in order to generate the output for the auspice visualizations. Since our root entity is a sample set when you click "SELECT DATA" click on the set of samples that you just ran Augur\_Prep on.

And in INPUTS section, assembly\_fastas attribute is set to "this.augur\_prep\_samples.assembly\_fastas", Remember it will look for **set of samples** not just sample, so **name the attribute to root entity and set it to plural**.

The rest of the inputs are:

- Build\_name in string format: Here we named it "B.1.147\_samples\_set", but you can call it whatever you like.
- sample\_Metadata.tsvs (Generated by the Augur\_Prep in the previous step) as "this.augur\_prep\_samples.augur\_metadata. **Again note the plural form!**

If you don't use the plural form then you will get an error see section 8.2 for details.

And it should look like this.

The screenshot shows the Terra Workflow Inputs interface for the 'Titan\_Augur\_Run' workflow. The 'Version' dropdown is set to 'v1.4.4'. Under 'Step 1', 'Select root entity type' is set to 'augur\_prep\_sample'. Under 'Step 2', 'SELECT DATA' shows '1 selected augur\_prep\_sample\_set'. In the 'augur\_prep\_sample' table, the attribute 'assembly\_fastas' is set to 'B.1.147\_all\_samples\_set'. In the 'augur\_prep\_sample\_set' table, the attribute 'sample\_metadata\_tsvs' is set to 'this.augur\_prep\_samples.augur\_metadata'.

Once your input form is complete, move on to the OUTPUTS form and select "Use Defaults". Terra will then populate the OUTPUTS form with all of the default outputs options generated by the workflow. If you forget to do this you won't have easily accessible results! **Save these changes by clicking the 'Save' button**

Once your INPUTS and OUTPUTS forms are complete, click the 'Save' button on the top right-hand side of the page. The yellow caution icons should disappear and the Run Analysis option should be made available

You are now ready to run the Augur\_Run workflow! Click on the 'Run Analysis' button to the right of the 'Outputs' tab. A popup window should appear titled 'Confirm launch'. If the 'Run Analysis' button is greyed out, you need to save your recent changes by clicking the 'Save' button.

Clicking the 'Launch' button should bring you to the 'Job History' panel where each sample will be queued for the Titan\_Augur\_Run analysis. The status will change from Queued -> Submitted -> Running

Video showing how to submit the Augur\_Run workflow.

View and Download the Augur workflow output reports

- 5 First, **verify the sample set has successfully completed** by looking at the 'Workflow Status' section in the top left of the 'Job History' panel. The job has completed when the sample set have a status of 'succeeded' with a green checkmark.

Since augur\_run analyze with set of samples together, the job status would appear this way, it completed analysis successfully (hence the green checkmark).

DASHBOARD DATA NOTEBOOKS WORKFLOWS **JOB HISTORY**

Submission (click for details) Data Entity No. of Workflows Status Actions Submission ID

Submission 1f6e1d8-aef9-4b6b-92c5-9d7091eaeb00 Data entity: Jan\_Augur\_Run Status: Done Submitted: Jul 8 2021 5:48:40 AM Submission ID: 1f6e1d8-aef9-4b6b-92c5-9d7091eaeb00

### Job Status

If you want to check what steps have finished on a job (can be helpful for troubleshooting), then click the job description under the submission column. This will take you to a page with more details on the job run.

DASHBOARD DATA NOTEBOOKS WORKFLOWS **JOB HISTORY**

Submission (click for details) Data Entity No. of Workflows Status Actions Submission ID

Submission 1f6e1d8-aef9-4b6b-92c5-9d7091eaeb00 Data entity: Jan\_Augur\_Run Status: Done Submitted: Jul 8 2021 5:48:40 AM Submission ID: 1f6e1d8-aef9-4b6b-92c5-9d7091eaeb00

Workflow Status: ✓ Successed 1 Workflow Configuration: Jan\_Augur\_Run

Data Entity: Jan\_Augur\_Run Submited by: Jan\_Augur\_Run Total Cost: N/A

Workflow ID: 1f6e1d8-aef9-4b6b-92c5-9d7091eaeb00 Call Caching: Enabled

Delete Intermediate Outputs: Close this Use Reference Disk: Close this

Search: Search Completion status: ▾

Data Entity	Last Changed	Status	Run Date	Messages	Workflow ID	Links
Jan_Augur_Run_2021-07-01T17-38-35 (augur_prep_sample_set)	Jul 8, 2021, 5:49 AM	✓ Successed	N/A	236+132+162+42+98+174+2150+1	1f6e1d8-aef9-4b6b-92c5-9d7091eaeb00	

Next click on the "Workflow Dashboard" icon on the far right under the links column. On the next page you will be able to tell what steps in the pipeline have successfully finished. Here all 3 tasks that were call finished successfully (hence the green checkmark).

Workflow Status: ✓ Successed

Workflow Timing: Start: Jul 9, 2021, 10:33 AM End: Jul 9, 2021, 10:46 AM

Links: Job Manager Execution Directory View execution log

Total Call Status Counts: ✓ 3 Succeeded

Call Lists:

- > titan\_augur\_run.version\_capture × 1
- > titan\_augur\_run.xarraycov2.nextstrain × 1
- > titan\_augur\_run.exp\_dists × 1
- > Submitted workflow script

Once confirming the job completed, go to the 'DATA' panel and under the 'TABLES' tab click on the sample table that you created and uploaded in step 4. It will be named 'sample (#)' which has the set of samples/entities in your file

DASHBOARD DATA NOTEBOOKS WORKFLOWS JOB HISTORY

TABLES DOWNLOAD ALL ROWS 0 rows selected

ONT_sample	(20)	gur_prep_sample.set_id	1	augur_prep_samples	asupice_input.json	combined_assemblies	keep_list	MAFFT	⋮
ONT_sample_set	(2)	an_Augur_Run_2021-07-01T17-3...	20 entities	asupice_B1147.all_samples.set.au...	all_samples_combined_assembly.fil...	all_samples_aligned.fasta_aligned.txt	all_samples_all	⋮	⋮
assembly_sample	(9)								⋮
assembly_sample_set	(1)								⋮
augur_prep_sample	(20)								⋮
<b>augur_prep_sample_set</b>	<b>(1)</b>								⋮
augur_sample	(1)								⋮

Augur\_Run output with added results after the analysis

Now the Terra sample table will have the additional attributes that were added by the workflow when you specified the output names (set to default in this example). You can reduce the number of fields you want to visualize by clicking the "gear" icon in top row on the right. Select only the fields you want to see then click "Done".

DASHBOARD DATA NOTEBOOKS WORKFLOWS JOB HISTORY

TABLES DOWNLOAD ALL ROWS 0 rows selected

ONT_sample	(20)	gur_prep_sample.set_id	1	augur_prep_samples	asupice_input.json	combined_assemblies	keep_list	MAFFT	⋮
ONT_sample_set	(2)	an_Augur_Run_2021-07-01T17-3...	20 entities	asupice_B1147.all_samples.set.au...	all_samples_combined_assembly.fil...	all_samples_aligned.fasta_aligned.txt	all_samples_all	⋮	⋮
assembly_sample	(9)								⋮
assembly_sample_set	(1)								⋮
augur_prep_sample	(20)								⋮
<b>augur_prep_sample_set</b>	<b>(1)</b>								⋮
augur_sample	(1)								⋮

Pop up menu with options of what metrics you want to see

The three files that are used for visualization are:

**auspice\_input\_json** - this output is generated from the NextClade analysis that is part of the Augur workflow. This file includes the samples for clade typing and the single sample placed on the tree. Downloading this file won't be saved to your local folders but will be opened in a browser. Make sure you right click on the page and "save as" it to your local directory

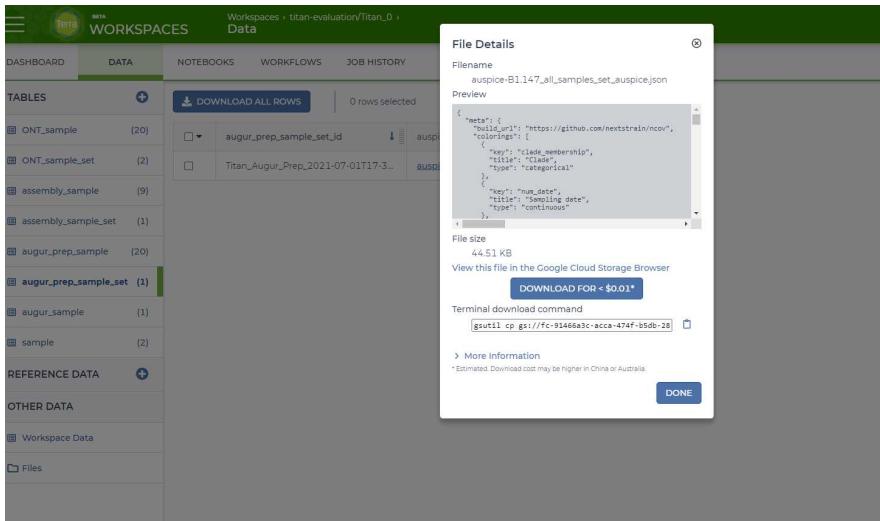
**combined\_assemblies** - Is the concatenation of all of the assemblies that were included in this phylogenetic analysis. For this examples it includes all of 20 samples combined into one single filtered fasta file combined\_assemblies.

**metadata\_merged** - Every metadata file from each sample that is generated by the augur\_prep step is merged into one single metadata file.

The auspice and combined assemblies determined by the pipeline can be viewed in the Terra sample report by scrolling to the right (or just only selecting those columns using the "gear" icon). You can download or copy this report by using either the 'Download All Rows' or 'Copy Page To Clipboard' buttons at the top of the table. To download a particular output file, click on the link under the column name a popup window titled 'File Details' should appear.

The 3 major files you want

Click the 'Download For < \$0.XX' button, which will download the consensus sequence to your local directory. This can be uploaded to auspice to visualize if desired.



Let's go ahead and download these 3 files to do some phylogenetic visualization in the next step.

Video showing how to download and save the output files.

#### Auspice Visualization

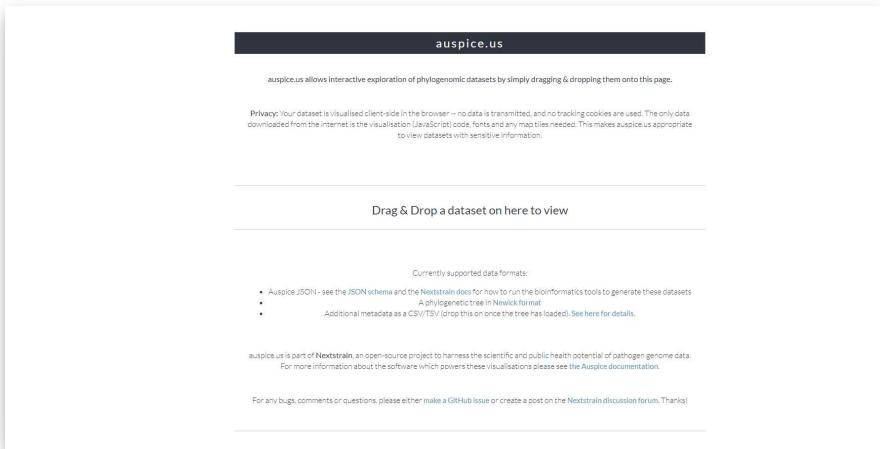
- 6 At the end of the Augur analysis, you would have a complete metrics of the genome assemblies, which can be used to analyze the relatedness of the samples via phylogenetic inference using Auspice.

Auspice will take a json file and build a tree that is visualized with NextStrain. The [CDC has genomic toolkit modules](#) that will probably be helpful in understanding the output we generate.

Module 3.2: Getting started with Nextstrain

Module 3.4: Walking through Nextstrain trees

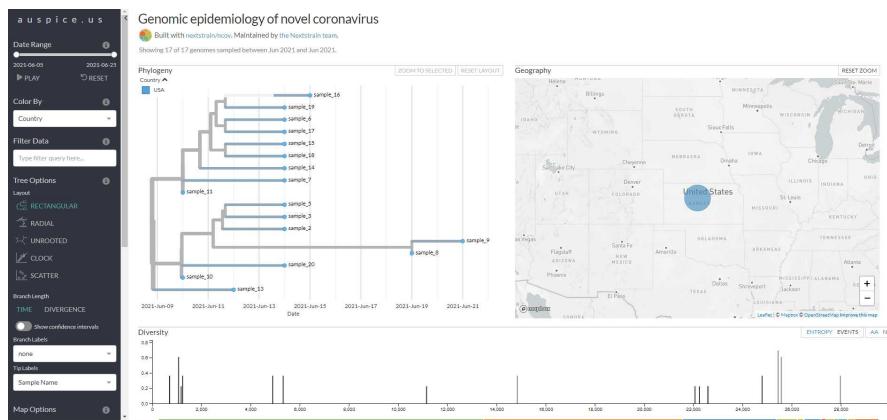
You should have downloaded auspice\_input\_json and metadata merged files from step 5. We will now use them to generate a phylogenetic visualization using [auspice.us](#). Go ahead and launch [auspice.us](#), it will land you directly on to the home page below.



Auspice.us home page

You can directly drag and drop the auspice\_input\_json file, for it to build the phylogenetic tree, and the output would

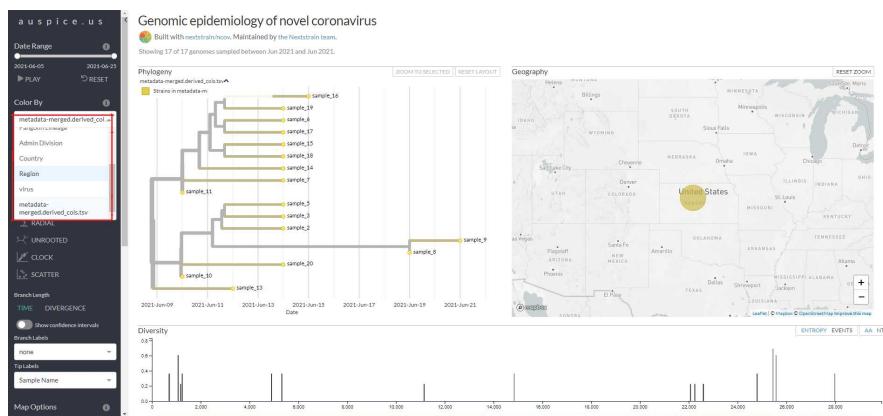
look something like this:



Auspice Visual

Hover over the tree and geographical map to see the divergence of the strains, toggle between the options and parameters based on your needs.

And if you wish to add metadata to observe any local outbreaks and quickly identify clusters within the samples that you analyzed, drag and drop the metadata\_merged file on the built tree.



After adding Metadata

**Note:** Once you drop the metadata file, all the attributes will show up in the "Color By" section (like in the picture above) where you can visualize your samples by whatever attribute that you have in your metadata file.

Visualization gives few default errors and it would not be a problem, but most importantly it gives the option to color code your samples by metadata variables.

**Since the metadata in this example is random dummy data so results are not accurate, but if you upload your metadata, it would allow you to 'Color By' the metadata variables.**

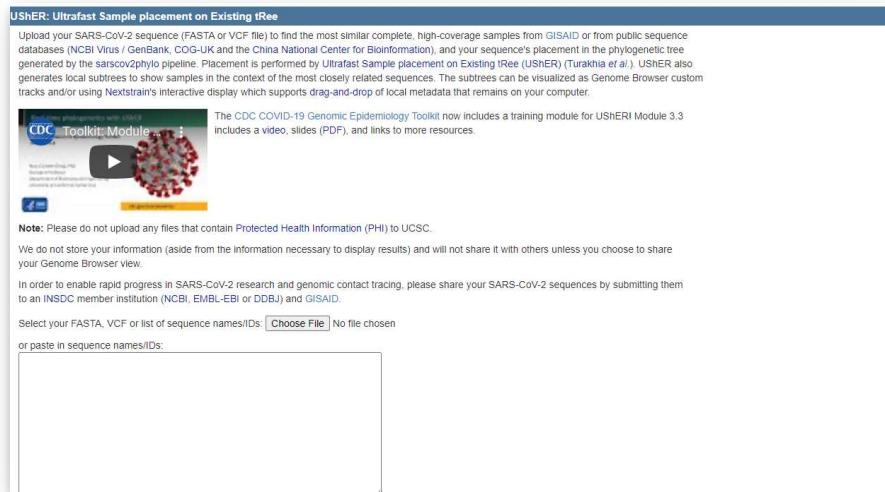
This video shows how to drop and visualize on auspice.

#### NextClade Visualization

- 7 You might want to look at a comparison to Auspice by visualizing through UCSC UShER too.

If you are not familiar with UShER see the [CDC's Genomic Toolkit Module 3.3](#) for more details. Additionally, there is a [preprint](#) for UShER as well as a [demo](#) released by the authors.

Start by navigating to the [UCSC UShER](#) tool home page.



UShER home page

Upload your concatenated fasta file (combined\_assembly from step 5) by clicking "Choose File" and selecting the combined\_assembly file then click "Upload". You can open the combined\_assembly file, copy and paste the sequence names/IDs, but uploading this way is easier.

UShER is quick and it will build phylogenetic placement against genome in [GISAID](#) or GenBank in few minutes. After uploading your assembly file pick which phylogenetic tree version you want to use from the drop down menu. **These are updated over time as new genomes are added so its a good idea to take note of the version you choose to use.**

UShER outputs the subtress results of 50 most closely related samples to your input samples.

You can pick the Phylogenetic tree version if you want to see the from where/how your samples are diverged, UShER would reproduce the results adding your samples to the existing tree nodes. We recommend using the top, most up to date tree.

Once UShER finishes, it will have constructed a 50 node tree based on the closest neighbor. The subtrees that are generated by the tool depends on divergence of the strains, if the strains are more divergent, it would show more subtrees, and if you wish to add more samples, UShER would add them to the existing tree based on sample divergence and mutation. You can visualize and explore this using the Nextstrain platform option.

---

The results from UShER would be similar to that of the Auspice tool, however UShER conveniently gives extra information like QC and other metrics which you see in table format on the UShER page.

If you wish to see the subtrees, select the options that is available below the table, it will launch the phylogenetic tree being rendered on Nextstrain platform. If you wish to display the metadata, you can drop the metadata on to the Nextstrain just like we did on Auspice (in step 6).

For this demo UShER took the concatenated assembly fastas of 20 samples and constructed a phylogenetic tree against all GISAID trees that are being maintained by the UShER and it gave 16 subtrees (See "Subtree Number" column).

If you hover over the "?" of the column variables/sample rows, a gray dialogue box would appear (like in the picture) showing the info on that attribute.

Downloads   Global phylogenetic tree with your sequences   TSV summary of sequences and placements   TSV summary of Spike mutations   ZIP file of subtree JSON and Nextclade files														
Pasta Sequence	Size (M)	nt	#Mixed bases aligned (?)	Inserted bases (?)	Deleted bases (?)	#SNVs used for placement (?)	#Masked SNVs (?)	Nextstrain class (?)	Neighboring sample in tree (?)	Lineage of neighbor (?)	#Imputed values for mixed bases (?)	#Maximally parsimonious placements (?)	Parsimony score (?)	Subtree number (?)
sample_10	29792	514	0	29988 (?)	0	0	4 (?)	0	20C	Walessi/HWIC-2960f/2020   2020-04-09	0	0	2 (view in Nextstrain)	
sample_11	29792	0 (?)	0	29782 (?)	0	0	8 (?)	0	20A	IHU/COV-0483   LR794668.1   ?	B.1.147	0	1	3 (view in Nextstrain)
sample_15	29773	0 (?)	0	29773 (?)	0	9 (?)	9 (?)	0	20A	USA/MOHHS-SC2014B/2020   MT49573.1   2020-03-24	B.1	0	1	3 (view in Nextstrain)
sample_14	29792	0 (?)	0	29782 (?)	0	0	8 (?)	0	20A	IHU/COV-0483   LR794668.1   ?	B.1.147	0	1	2 (view in Nextstrain)
sample_15	29792	1695	0	27985 (?)	bases 55 - 29838 align to reference bases 05 - 29305	6 (?)	0	0	20A	IHU/COV-0483   LR794668.1   ?	B.1.147	0	2	0 (view in Nextstrain)
sample_16	29792	0 (?)	0	29782 (?)	0	0	8 (?)	0	20A	IHU/COV-0483   LR794668.1   ?	B.1.147	0	1	0 (view in Nextstrain)
sample_17	29792	0 (?)	0	29782 (?)	0	0	6 (?)	0	20A	IHU/COV-0483   LR794668.1   ?	B.1.147	0	1	7 (view in Nextstrain)
sample_18	29794	108	0	29405 (?)	0	0	8 (?)	0	20A	IHU/COV-0483   LR794668.1   ?	B.1.147	0	0	6 (view in Nextstrain)
sample_19	29792	505	0	29286 (?)	0	0	6 (?)	0	20A	IHU/COV-0483   LR794668.1   ?	B.1.147	0	2	0 (view in Nextstrain)
sample_2	29792	0 (?)	0	29773 (?)	0	0	9 (?)	7 (?)	20C	USA/UT-UPHL-20121082/2020   MV210972.1   2020-06-19	B.1.446	0	20	11 (view in Nextstrain)
sample_20	29792	2346	0	27426 (?)	0	0	9 (?)	0	20A	Engeland/LIVE-0E969/2020   2020-04-11	g	0	0	12 (view in Nextstrain)
sample_3	29792	2148	0	27608 (?)	0	0	8 (?)	0	20C	USA/UT-UPHL-20121082/2020   MV210972.1   2020-06-19	B.1.446	0	255	0 (view in Nextstrain)
sample_6	29792	2144	0	27626 (?)	0	0	6 (?)	0	20C	USA/UT-UPHL-20121082/2020   MV210972.1   2020-06-19	B.1.446	0	235	0 (view in Nextstrain)
sample_6	29792	0 (?)	0	29782 (?)	0	0	8 (?)	0	20A	IHU/COV-0483   LR794668.1   ?	B.1.147	0	1	0 (view in Nextstrain)
sample_7	29792	0 (NP)	0	29780 (NP)	0	0	4 (?)	0	20A	IHU/COV-0483   LR794668.1   ?	R.1.147	0	4	4 (view in Nextstrain)

Output metrics generated by UShER tool

Colors are represented to show the variations of each attribute between samples.

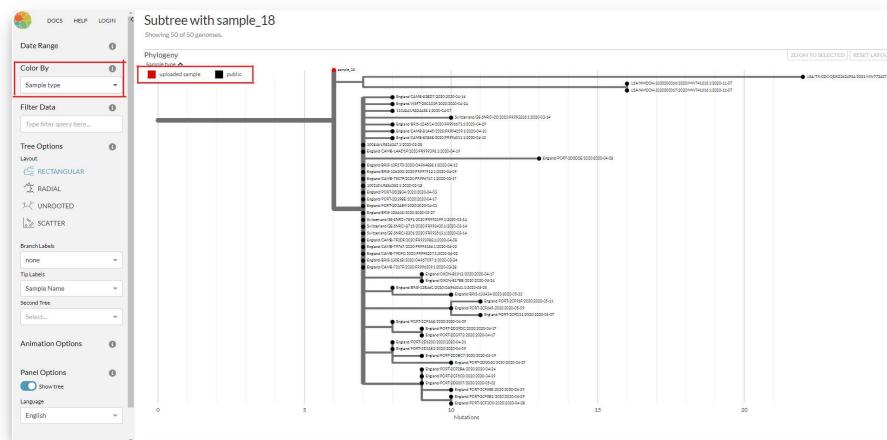
The lower the score the lighter the color, which means you samples are of high quality when it is green. When the score is high the output would appear in Red. In general the higher the score/red in color in the metrics, suggests that the sample sequence has many errors and sample placements might not be reliable.

**As an example:** In the image above the Parsimony score, which explains the number of mutations on the branch, for Sample\_10 is highlighted in red, this says the sample has relatively more mutations added to its branch than the others.

Refer to [CDC's Genomic Toolkit Module 3.3](#) for more information on this.

If you want to see the subtrees click the subtree section of the row you are interested in and the link associated to that sample will redirect to the subtree visual rendered on the Nextstrain platform. Nextstrain distinguishes the public samples in Black and your input samples in Red. Subtrees basically shows how your sample is diverged and placed from the public samples.

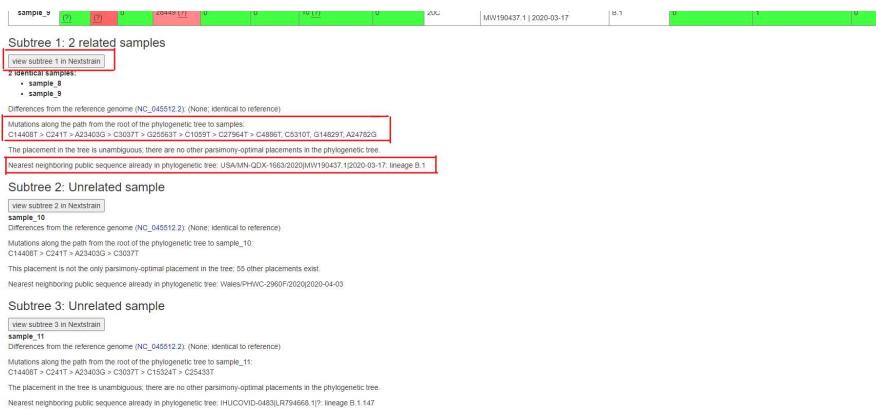
Here is an example subtree:



To view the subtree, sample\_18 was selected and opted to color by sample type from the menu on left side. Nextstrain distinguishes the public samples in Black and your input samples in Red.

You can also see the subtress by scrolling down the page and selecting from the individual subtree.

Alongside, you will find an extra information such as the path of the mutation from root of the tree to your sample, most closely related sample in the global tree and its pango lineage that has been assigned to the sample in tree.



## How to debug a failed run

8

### 1. Failure due to the unavailability of the compute environment.

Augur\_run workflow script requests 78 CPUs in order to generate the auspice and nextstrain build. If the analysis fails due to lack of resources, requesting for more CPUs to Terra would be the solution.

The screenshot shows a 'Messages' panel with the following log entry:

```

Failures in titan_augur_run.sarscov2_nextstrain / index -1 / attempt 1
▼ [
  ▼ 0 : {
    "message" : "Workflow failed"
    ▼ "causedBy" : [
      ▼ 0 : {
        "message" :
          "Task sarscov2_nextstrain.mafft:NA:1 failed. The job
          was stopped before the command finished. PAPI error
          code 9. Execution failed: allocating: selecting
          resources: selecting region and zone: no available
          zones: us-central1: 78 CPUS (17/24 available) quota too
          low"
      }
    ]
  }
]

```

For more details on how to fix this see [step 2 of our Terra Troubleshooting protocol](#).

### 2. Failure for not specifying the input attributes correctly.

There are two flavors of this error for Augur\_Run:

- Singular instead of plural notation
- Using sample instead of Sample\_Set

First, the singular instead of plural problem:

Augur\_run workflow looks for concatenated set of fasta files, **you need to set the root entity to "name\_sample\_set" (with name being specific to your dataset) and the input attribute notation of "this.name\_samples.assembly\_fasta and metadata as "this.name\_samples.metadata"** rather than using "this.assembly.fasta" and "this.metadata" as we did with the Augur\_prep protocol.

In other words, **you select the name\_sample\_set (singular) in root entity, however for the input use name\_samples (plural) notation**, since it is looking for all the files as concatenated set.

You will see this error if you didn't use the plural:

DASHBOARD DATA NOTEBOOKS WORKFLOWS **JOB HISTORY**

← Job History > d8815bd-d32d-41c0-9e51-4c7aaa74eb19 - Workflow 1ec376a8-d348-4755-a80e-d34a5c962275

Workflow metadata fetched in 423ms

**Workflow Status**  
⚠ Failed

**Workflow Timing**  
Start: Jul 8, 2021, 5:02 PM  
End: Jul 8, 2021, 5:02 PM

**Links**  
[Job Manager](#) [Execution Directory](#) [View execution log](#)

Workflow-Level Failures

```
[{"error": {"message": "Workflow input processing failed", "causedBy": [{"error": {"message": "Failed to evaluate input 'assembly_fastas' (reason 1 of 1): An Array[File]+ must contain at least one element"}}, {"error": {"message": "Failed to evaluate input 'sample_metadata_tsv' (reason 1 of 1): An Array[File]+ must contain at least one element"}]}]}
```

Calls

Total Call Status Counts  
No calls have been started by this workflow.

Submitted workflow script

Error when you use singular instead of plural in inputs

Second, we will look at the problem of using sample instead of sample set as your root entity type:

Dashboard Data Notebooks Workflows Job History

Job History • Submission bcd29224-bd8a-40fe-b120-2e0290000001 • Workflow 538a3702-9641-4487-a505-d3d497540000

Workflow metadatas 80207 (n=24) mg

Workflow Status ⚠ Failed

Workflow Level Failures ⚠ Failed

```
1 |   v = {
2 |     "message": "Workflow failed"
3 |   }
4 |   if (v.message === "Workflow failed") {
5 |     return v;
6 |   }
7 |   else {
8 |     throw new Error(`Workflow ${v.message} failed`);
9 |   }
10 | }
```

Workflow Timing

Start	End
Jul 8, 2021, 3:59 PM	Jul 8, 2021, 4:15 PM

Links

[Job Manager](#) [Execution Directory](#) [View execution log](#)

Call Status

Total Call Status Counts

✓ 1 Succeeded

⚠ 1 Failed

Call Log

```
v titan_wmqr_run_version_capture × 1
v titan_wmqr_run_arxiv0_pextestrain × 1
```

Name	Attempt	Status	Start	End	Call Caching Result	Links
N/A	1	<span style="color: red;">⚠ Failed</span>	Jul 8, 2021, 3:59 PM	Jul 8, 2021, 4:15 PM		<span style="color: orange;">⚠ Message</span>

## Error in Terra.bio job history

If you actually go into the draft\_augur\_tree.log file in the GCP you will find this error

draft augur tree.log output

The solution here is to make sure your root entity type is set to the sample set and not sample.

Make sure you are setting the root entity type to sample\_set!!!

### **3. Error due to spacing in collection name**

In this example of a failed run we have an error caused by adding spaces into your collection name. If there is a failure for some other reason you can follow similar steps to understand why your job failed. Here we will be using Illumina paired-end data, but the process of debugging an error is the same if you have ONT data.

If a run fails you will see this indicated in the job history screen in the "status" column.

The screenshot shows the Data & Tools Workflow Dashboard. At the top, there's a navigation bar with tabs for Dashboards, Data, Notebooks, Workflows, and Job History. The Job History tab is selected, showing a list of recent submissions. One submission is highlighted in red, indicating it failed. The submission details are as follows:

Workflow Status	Submitted by	Total Run Cost
⚠ Failed: 1	ipa@ipa.gov	N/A
Workflow Configuration ipa-team-test@ipa.iiumina.PE	May 13, 2021, 5:40 PM	
Data Entity 3015780998_ZZYGIWY (sample)	Submission ID 97f5da29-71d1-45e8-9938-82ba0a36b788	Call Caching Disabled
Delete Intermediate Outputs Disabled	Use Reference Disks Disabled	

Below the table, there's a search bar and a dropdown menu for completion status. A message at the bottom right says "Workflow Dashboard [ipha]".

## Job Failure

To understand why it failed click on the "workflow dashboard" icon in the "links" column. This will take you to a new screen and you can click the arrows next to the "message" to expand the message and see what it says. Here we see there are two errors that direct us to a log file to check. To find out more click on the "execution directory" icon under the links header. This will take you to the google bucket with all the output from the run.

Workspaces : toast-learn-test/Job\_Failure\_Example

Job History

Dashboard Data Notebooks Workflows Job History 308 HISTORY

Job History • Submission 97f0da29-71d1-4e5b-9938-01a0a367b798 • Workflow e1687504-3421-4a20-9de1-11e35422d3e3

Workflow metadata fetched in 44ms

Workflow Status

Workflow-Level Failures □

1 "e": { "id": "1", "message": "Workflow Failed", "causesBy": { "0": [ { "id": "1", "message": "openFile Failed", "causesBy": { "0": [ { "id": "1", "message": "readFile Failed", "causesBy": [ { "id": "1", "message": "Read file failed" } ] } ] } ] } } }

Workflow Timing

Start: May 13, 2021, 5:41 PM  
End: May 13, 2021, 5:45 PM

Links

File Manager Execution Directory View execution log  
**Execution directory**

Calls

Total Call Duration: 0ms

## Job failure messages

Follow the file path in the google bucket to the log files that were referenced in the error messages (in red/orange text in

the above photo).

Google Cloud Platform		Select a project	Search products and resources						
Cloud Storage		< Bucket details							
		fc-650035cc-5b56-433b-95e6-27c1f5cfbb7e							
OBJECTS	CONFIGURATION	PERMISSIONS	RETENTION						
UPLOAD FILES	UPLOAD FOLDER	CREATE FOLDER	MANAGE HOLDS						
DELETE	DOWNLOAD								
Filter by name prefix only	Filter	Filter objects and folders	More						
<input type="checkbox"/> <a href="#">factory.log</a>	4 KB	text/plain; charset=UTF-8	May 13, 2021, 9:40:11 PM Standard	May 13, 2021, 9:45:11 PM	Not authorized	Google-managed key	—	None	
<input type="checkbox"/> <a href="#">gcs_deployment.sh</a>	4.9 KB	text/plain; charset=UTF-8	May 13, 2021, 9:41:28 PM Standard	May 13, 2021, 9:41:28 PM	Not authorized	Google-managed key	—	None	
<input type="checkbox"/> <a href="#">localization.sh</a>	1.1 KB	text/plain; charset=UTF-8	May 13, 2021, 9:41:28 PM Standard	May 13, 2021, 9:41:28 PM	Not authorized	Google-managed key	—	None	
<input type="checkbox"/> <a href="#">gcs_tracer.sh</a>	13.4 KB	text/plain; charset=UTF-8	May 13, 2021, 9:41:28 PM Standard	May 13, 2021, 9:41:28 PM	Not authorized	Google-managed key	—	None	
<input type="checkbox"/> <a href="#">pipeline/logo/</a>	—	Folder	—	—	—	—	—	—	
<input type="checkbox"/> <a href="#">rc</a>	2 KB	text/plain; charset=UTF-8	May 13, 2021, 9:40:29 PM Standard	May 13, 2021, 9:41:09 PM	Not authorized	Google-managed key	—	None	
<input type="checkbox"/> <a href="#">script</a>	1.9 KB	text/plain; charset=UTF-8	May 13, 2021, 9:41:28 PM Standard	May 13, 2021, 9:41:28 PM	Not authorized	Google-managed key	—	None	
<input type="checkbox"/> <a href="#">saber</a>	582 B	text/plain; charset=UTF-8	May 13, 2021, 9:45:23 PM Standard	May 13, 2021, 9:45:23 PM	Not authorized	Google-managed key	—	None	
<input type="checkbox"/> <a href="#">stale</a>	49 B	text/plain; charset=UTF-8	May 13, 2021, 9:45:24 PM Standard	May 13, 2021, 9:45:24 PM	Not authorized	Google-managed key	—	None	

Click on log file referenced in the error message.

Click on the "Authenticated URL" link that will take you to a text file.

Cloud Storage

Object details

Bucket: Buckets > hf-600300c5-5895-42b0-90e5-2719cfef50fe > 11976e297101-056f-9038-02b0-00000000768 > 35a\_luminar.ap\_ > e1f08703048421-4a02-90bf-11e9342203e9 > .cafr-read\_2c23fb0 > resJUC2JHm > hf-602205a8f010447-82237cd822779272 > cafr-testqz.m

Buckets > hf-600300c5-5895-42b0-90e5-2719cfef50fe > 11976e297101-056f-9038-02b0-00000000768 > 35a\_luminar.ap\_ > e1f08703048421-4a02-90bf-11e9342203e9 > .cafr-read\_2c23fb0 > resJUC2JHm > hf-602205a8f010447-82237cd822779272 > cafr-testqz.m

Browser Monitoring Settings

Since you are not authorized to know the public access status of this object, it is possible that the public URL displayed is not valid.

Overview

Type: test/plain; charset=UTF-8

Size: 4 kB

Created: May 13, 2021, 5:45:11 PM

Last modified: May 13, 2021, 5:45:11 PM

Custom time: Custom URL: [https://storage.googleapis.com/hf-600300c5-5895-42b0-90e5-2719cfef50fe/e1f08703048421-4a02-90bf-11e9342203e9/.cafr-read\\_2c23fb0/resJUC2JHm/hf-602205a8f010447-82237cd822779272/cafr-testqz.m](https://storage.googleapis.com/hf-600300c5-5895-42b0-90e5-2719cfef50fe/e1f08703048421-4a02-90bf-11e9342203e9/.cafr-read_2c23fb0/resJUC2JHm/hf-602205a8f010447-82237cd822779272/cafr-testqz.m)

Fancy view Fuzzy view Log viewer

Authenticated URL: [https://storage.cloud.google.com/hf-600300c5-5895-42b0-90e5-2719cfef50fe/e1f08703048421-4a02-90bf-11e9342203e9/.cafr-read\\_2c23fb0/resJUC2JHm/hf-602205a8f010447-82237cd822779272/cafr-testqz.m](https://storage.cloud.google.com/hf-600300c5-5895-42b0-90e5-2719cfef50fe/e1f08703048421-4a02-90bf-11e9342203e9/.cafr-read_2c23fb0/resJUC2JHm/hf-602205a8f010447-82237cd822779272/cafr-testqz.m)

gcsf URL: [gs://hf-600300c5-5895-42b0-90e5-2719cfef50fe/e1f08703048421-4a02-90bf-11e9342203e9/.cafr-read\\_2c23fb0/resJUC2JHm/hf-602205a8f010447-82237cd822779272/cafr-testqz.m](gs://hf-600300c5-5895-42b0-90e5-2719cfef50fe/e1f08703048421-4a02-90bf-11e9342203e9/.cafr-read_2c23fb0/resJUC2JHm/hf-602205a8f010447-82237cd822779272/cafr-testqz.m)

Permissions

Public access: Not authorized

Read access: None ✓

Metadata policy: None

Encryption type: None/managed key

## Authenticated URL link

In the text file we can see that there was an error that caused by there being a space between "Bad" and "Sample" in our file path that was created when we made "Bad Sample" rather than "Bad\_Sample" as our collection name.

fastqc\_raw.log file showing the error.

[Video of the whole process.](#)

These error messages can sometimes be hard to understand. Contact [TOAST@cdc.gov](mailto:TOAST@cdc.gov) if you need help debugging job failures.

**Citation:** Anusha Ginni, Jill V Hagey (07/28/2021). Auger/Auspice/UShER: SARS-CoV-2 Cluster Detection Workflow for the Terra Platform.  
<https://dx.doi.org/10.17504/protocols.io.bv4kn8uw>

This is an open access protocol distributed under the terms of the **Creative Commons Attribution License** (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.