2

Oct 18, 2021

Auger/Auspice/UShER SARS-CoV-2 Cluster Detection Workflow for the Terra Platform V.2

protocol

1

Anusha Ginni[1], Jill V Hagey[1], Michael Weigand[1], Technical Outreach and Assistance for States Team[1]

[1]Centers for Disease Control and Prevention

1

protocol .

Anusha Ginni

The Titan_Augur workflow processes concatenated assembly fasta data for SARS-CoV-2 phylogenetic analysis and cluster detection using Augur, the modular bioinformatics engine behind Nextstrain. Two workflows, Titan_Augur_Prep and Titan_Augur_Run, must be completed in series and produce the required inputs for visualization with Auspice to evaluate genetic relatedness among sequences or draw phylogenetic inference.

Inputs for Titan_Augur_Prep include the assembled fasta, metadata (uploaded tsv), and PANGO lineage assignments for each sample. Running Titan on Terra generates the necessary assembly fasta and PANGO lineage assignments, but these data can also be uploaded if prepared using another pipeline. Titan_Augur_Prep processes these data and outputs the correctly formatted metadata table and concatenated fasta files required for subsequent Titan_Augur_Run analyses. Titan_Augur_Run then generates the inputs needed for phylogenetic inference on the Auspice or Nextstrain interactive visualization tools.

This protocol also includes steps for visualization using Auspice as well as the UCSC UShER webportal.

For technical assistance, please contact: **TOAST@cdc.gov**

Anusha Ginni, Jill V Hagey, Michael Weigand, Technical Outreach and Assistance for States Team 2021. Auger/Auspice/UShER SARS-CoV-2 Cluster Detection Workflow for the Terra Platform. **protocols.io**
https://protocols.io/view/auger-auspice-usher-sars-cov-2-cluster-detection-w-bymepu3e
Technical Outreach and Assistance for States Team

(ILLUMINA MENU) Protocols for SARS-CoV-2 Library Prep with ARTIC Primers, Bioinformatic Analysis, and Database Submission

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

protocol ,

Sep 28, 2021

Oct 18, 2021

Oct 18, 2021 Anusha Ginni

53638

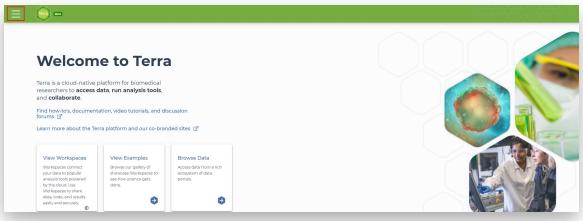
Part of collection
(ILLUMINA MENU) Protocols for SARS-CoV-2 Library Prep with ARTIC Primers, Bioinformatic Analysis, and Database Submission

:

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

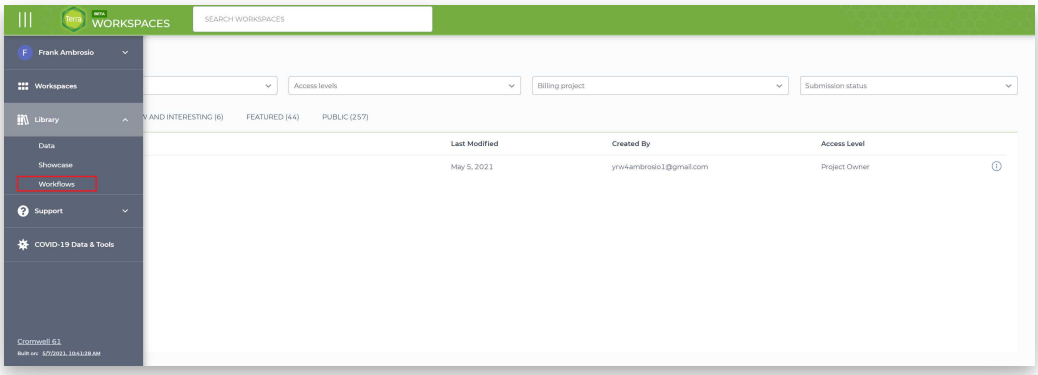
2

Importing the Titan_Augur Workflows from Dockstore to the Terra User Workspace

Note: *The Augur workflow includes two steps: Titan_Augur_Prep and Titan_Augur_Run.*

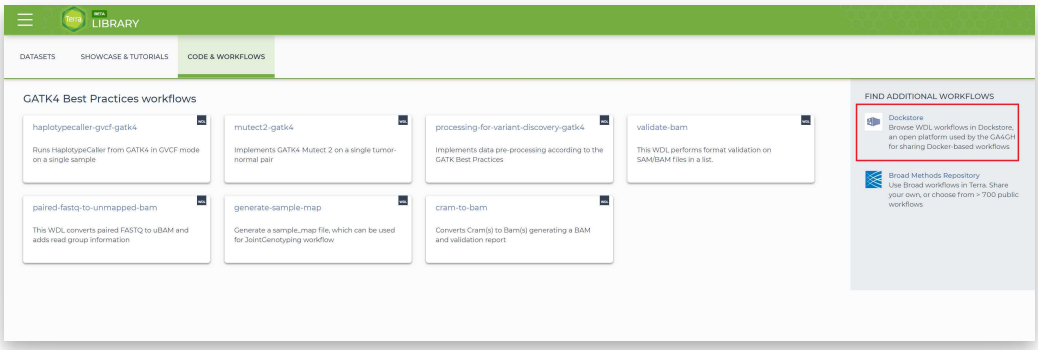
The Titan_Augur_Prep workflow prepares the metadata and assembly fasta required by the Titan_Augur_Run workflow, so both pipelines must be imported into your workspace.

The Augur workflows are hosted on the Dockstore repository and must 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.



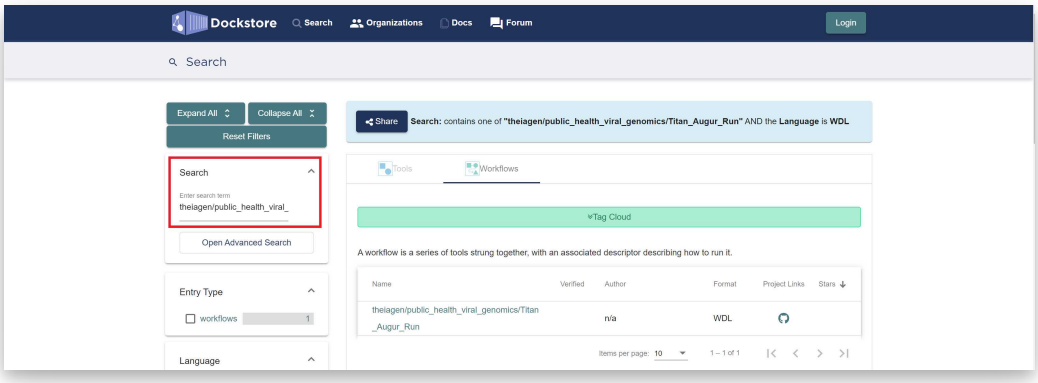
Find the 'Workflows' button listed under the 'Library' tab in the selection panel.

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



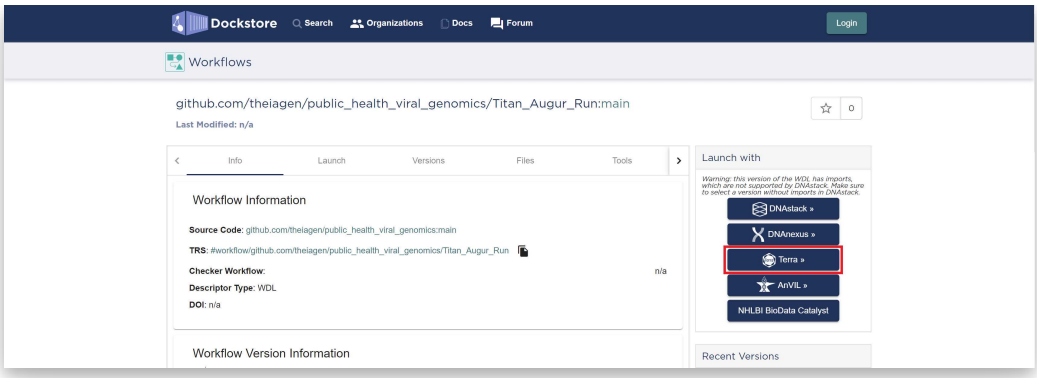
Workflows panel with link to Dockstore.

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

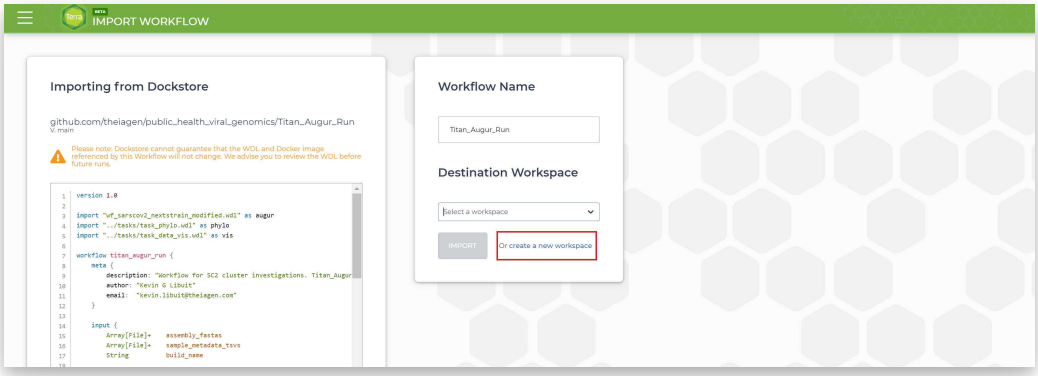
Click the 'theiagen/public_health_viral_genomics/Titan_Augur_Run' link. This will take you to a page where you can import the workflow into your Terra workspace.



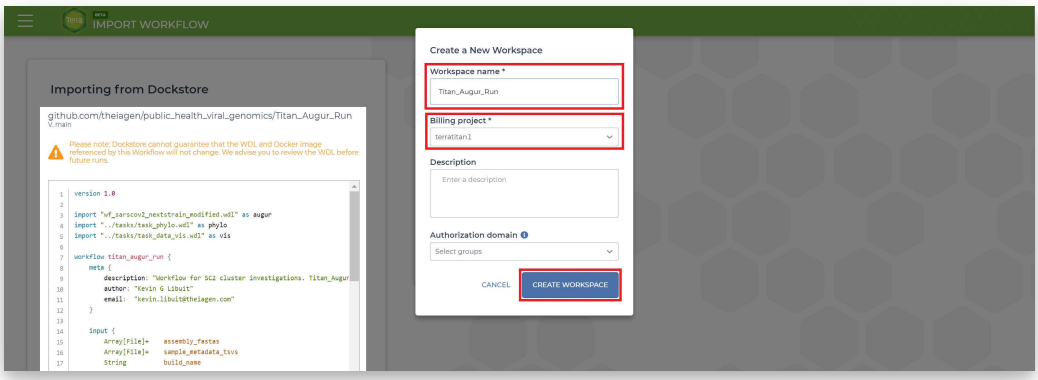
Dockstore link to terra

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.

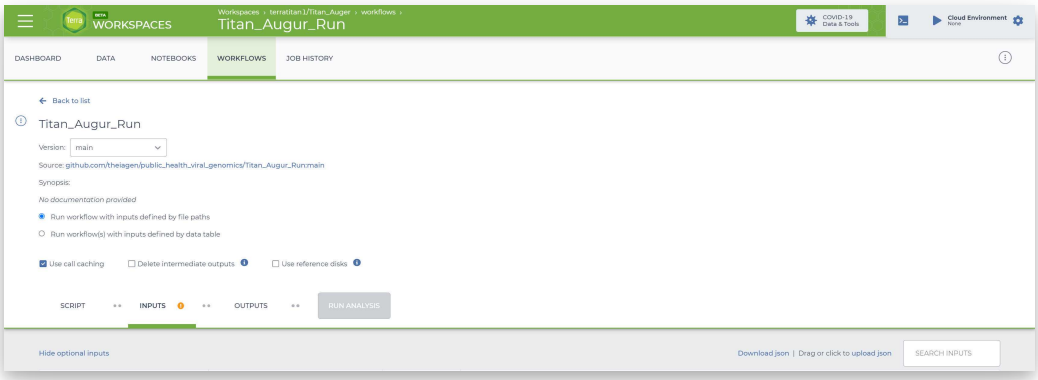


A pop-up window titled 'Create a New Workspace' should appear. Name your new workspace and associate it with a billing account using the 'Billing project' drop-down menu (this is the Terra Billing Account you created in the previous step should be available). Finally, click the 'Create Workspace' button.



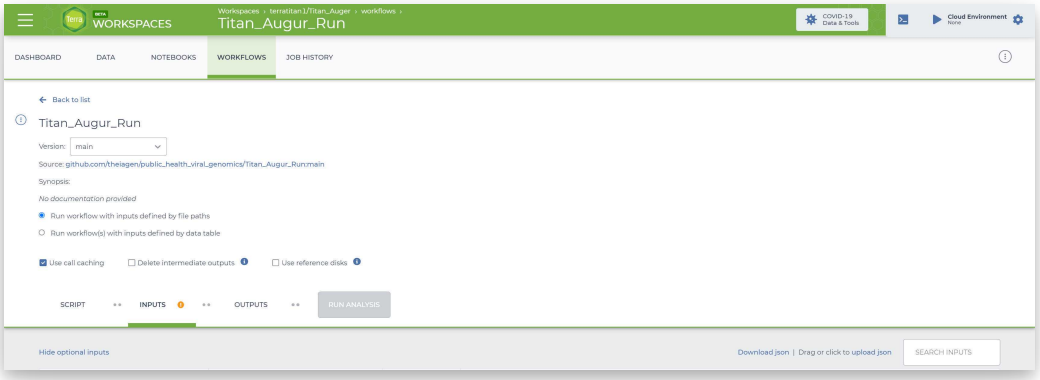
New Workspace Panel

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 complete the same steps to import the **theiagen/public_health_viral_genomics/Titan_Augur_Prep** workflow as well.

The video below demonstrates how to import both the Titan_Augur_Prep and Titan_Augur_Run workflows.

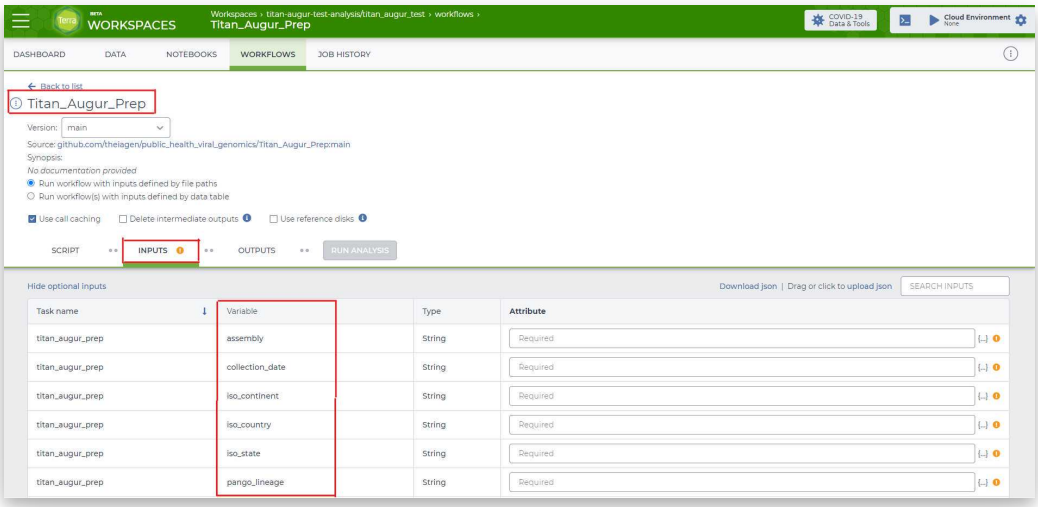
Analysis Inputs/Data uploading

3

Before you begin: Datasets with <15 samples must include some genetic diversity and varied

collection dates to ensure the Augur workflow runs successfully. This can be achieved by adding an outlier sample with a discrepant lineage and collection date.

The Augur workflows require the assembled fasta sequence (consensus sequence), PANGO linage assignments, and metadata for each sample. These required inputs are listed on the workflow page:



The Titan_Augur_Prep workflow page where required inputs can be added from the 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.

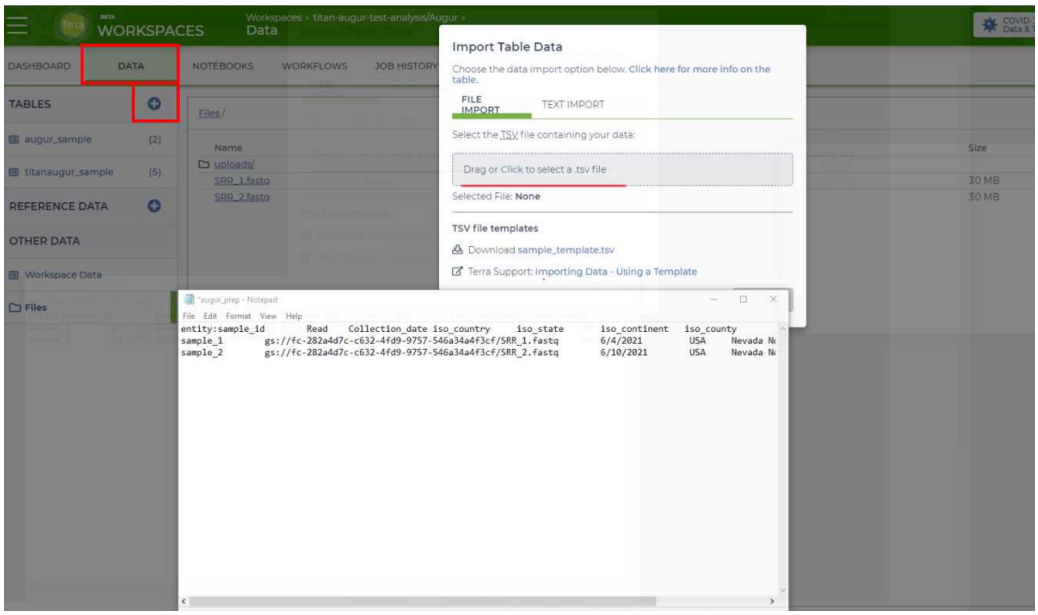
Construct a file with the following columns that are required metadata for each sample of interest:

- 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 assocaited 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 genome characterization on Terra and all the metadata is associated with your samples, skip to Step 4 to select the Titan_Augur_Prep and Titan_Augur_Run required inputs from the TABLE.

If you did not perform any previous genomic characterization of SARS-CoV-2 sequences on Terra, the assemblies and associated metadata file must first be uploaded to run the Augur workflows (see sub-steps below).

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

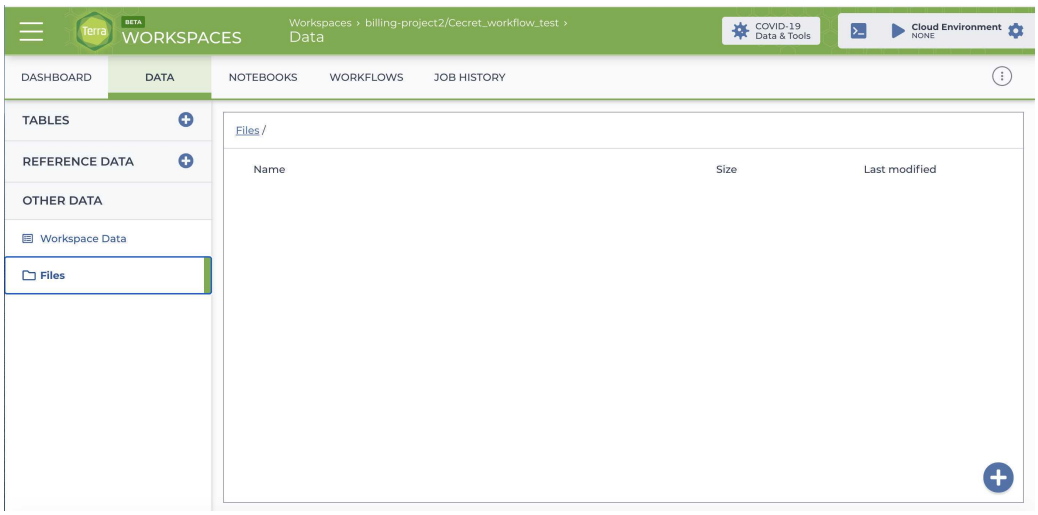
1. Upload on Terra (can only do single files).
2. Upload to Google bucket and link to Terra (can do single files or bulk files/folders).
3. Upload via 'https://app.terra.bio/#**upload**' -- **This is the easiest option**

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.

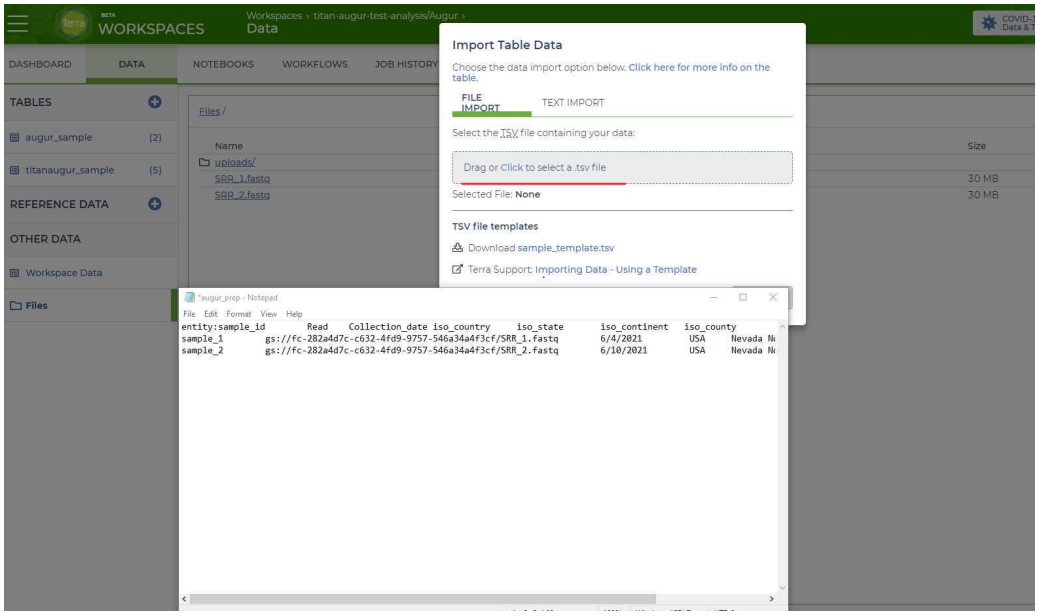


TABLE data updated with the file paths before uploading to workspace.

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
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		

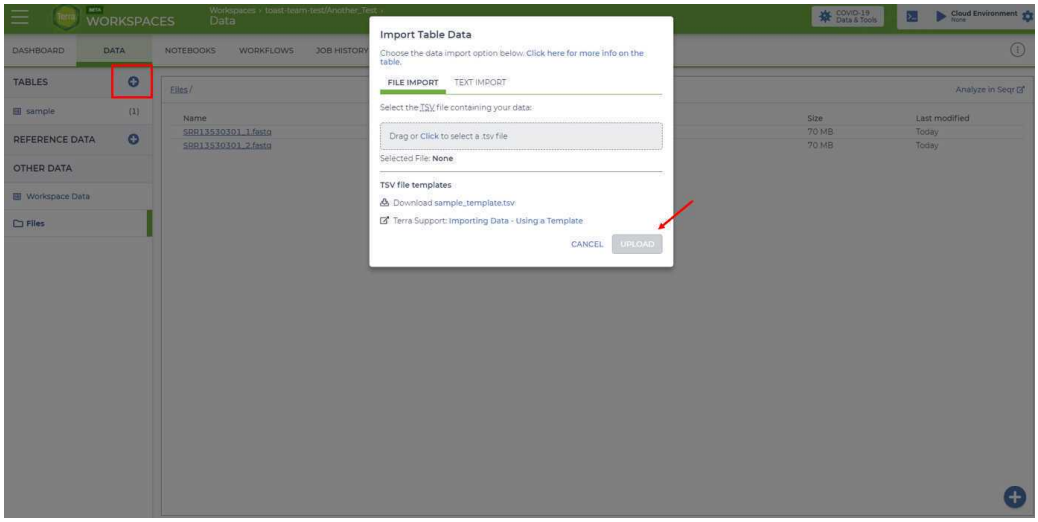
Example Terra sample file

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.)

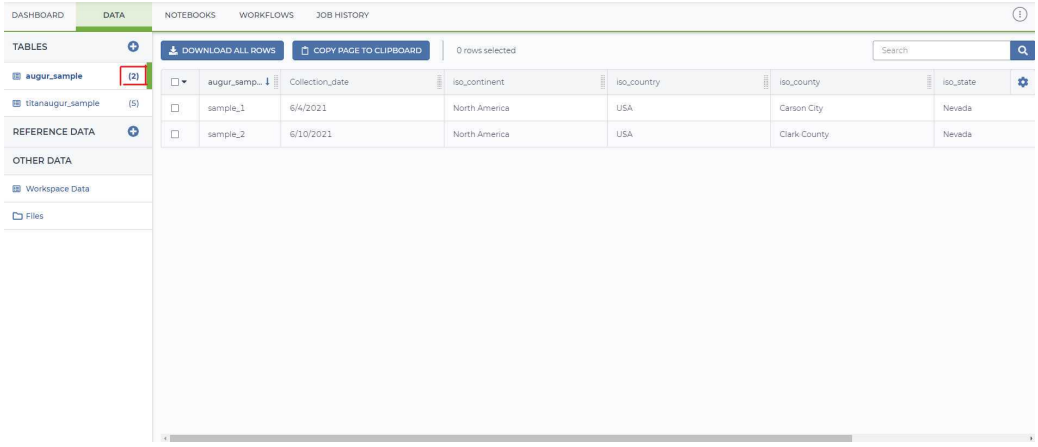
Finally, the completed Terra sample file will need to be uploaded to the workspace. On

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.



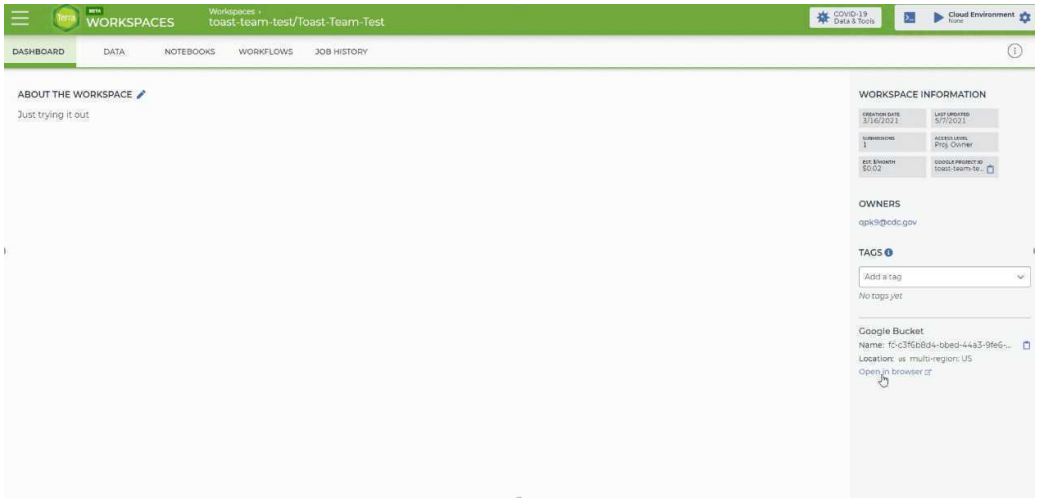
The workspace 'Data' panel after successfully uploading the fastq sequence files and TABLE file

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)

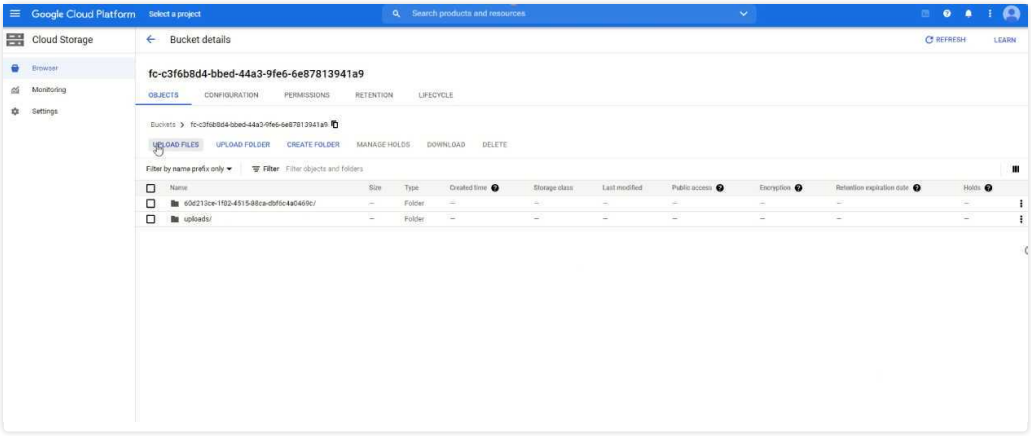
3.2 You will likely have many samples to upload and you can do this by going directly to your Google bucket.

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

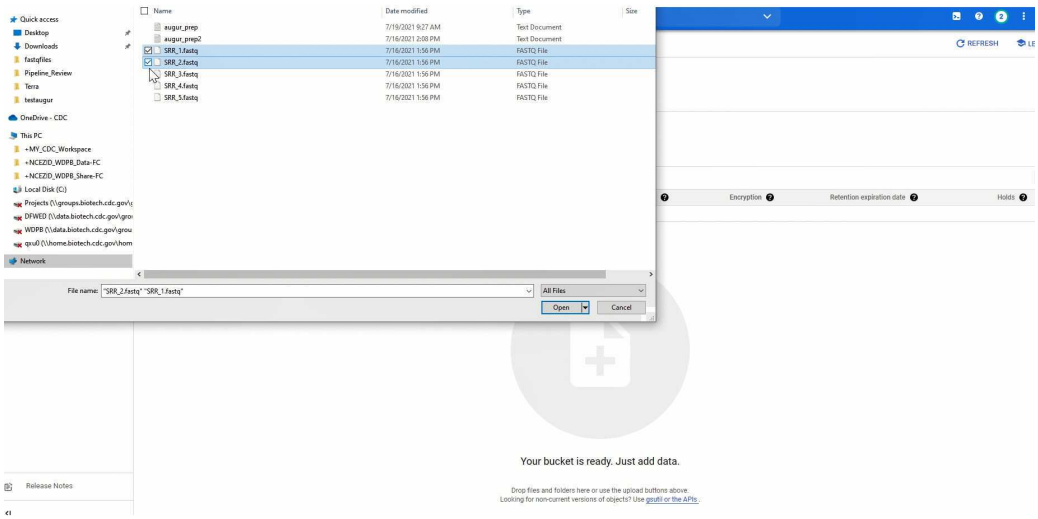


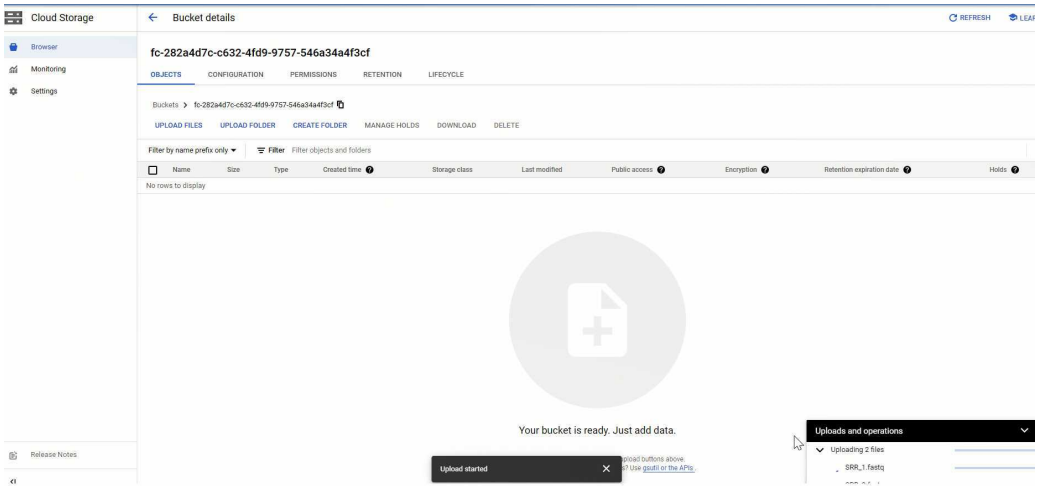
The Dashboard tab of your workspace.

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

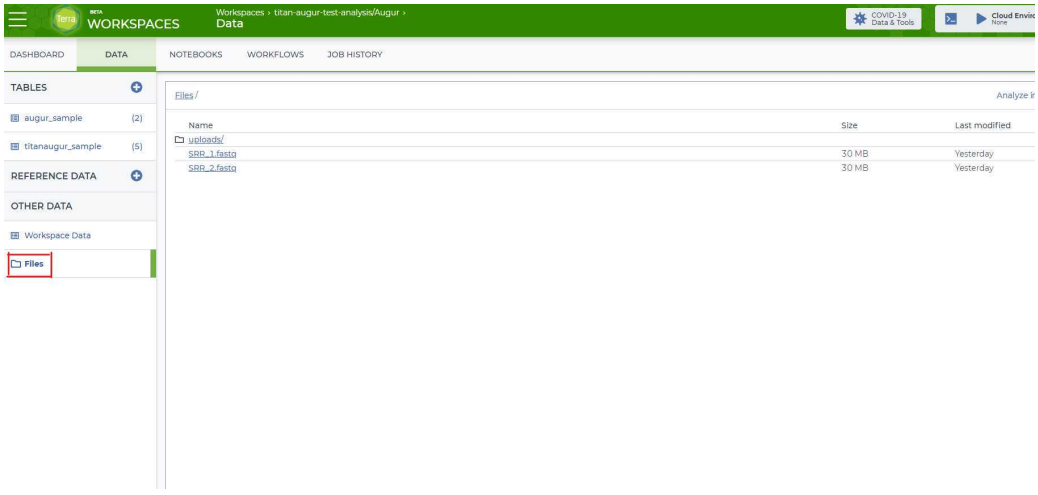


Click “UPLOAD FILES” in the middle of this page to upload single or multiple fastq files. Or you can click “UPLOAD FOLDER” to upload a folder with multiple fastq files stored inside.





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



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.

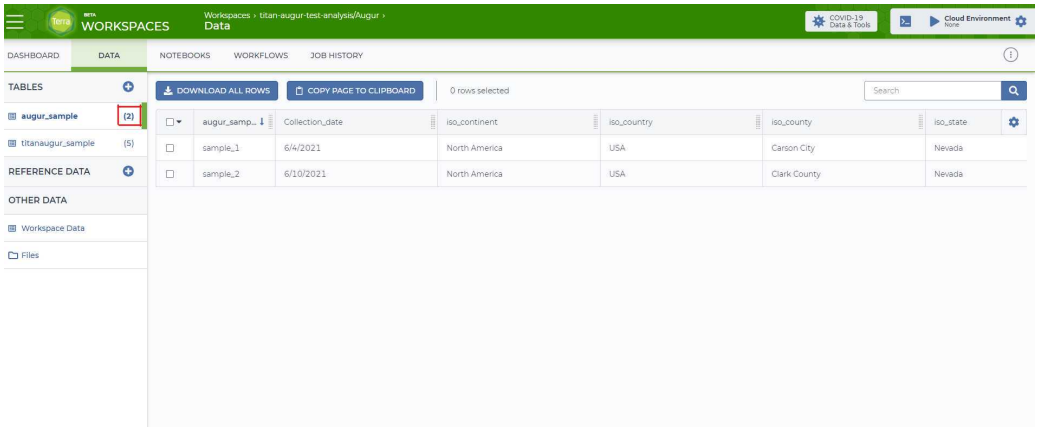


TABLE data showing metadata of the samples.

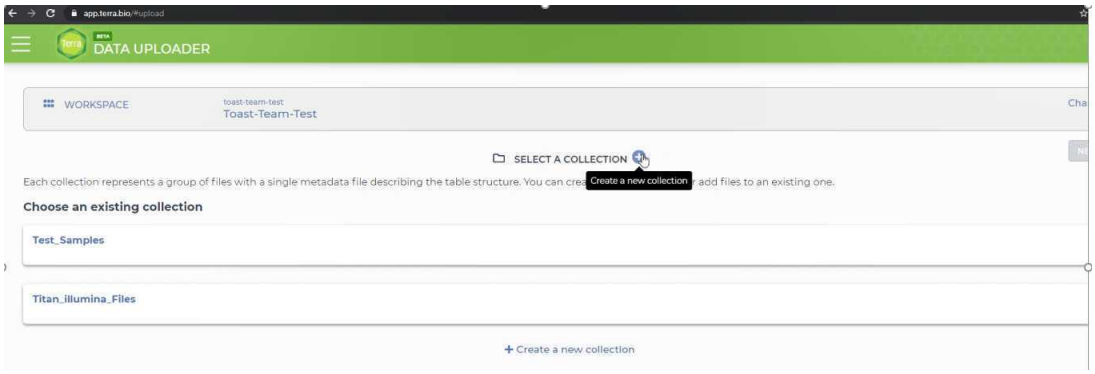
Here is a short video demonstration:

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**'

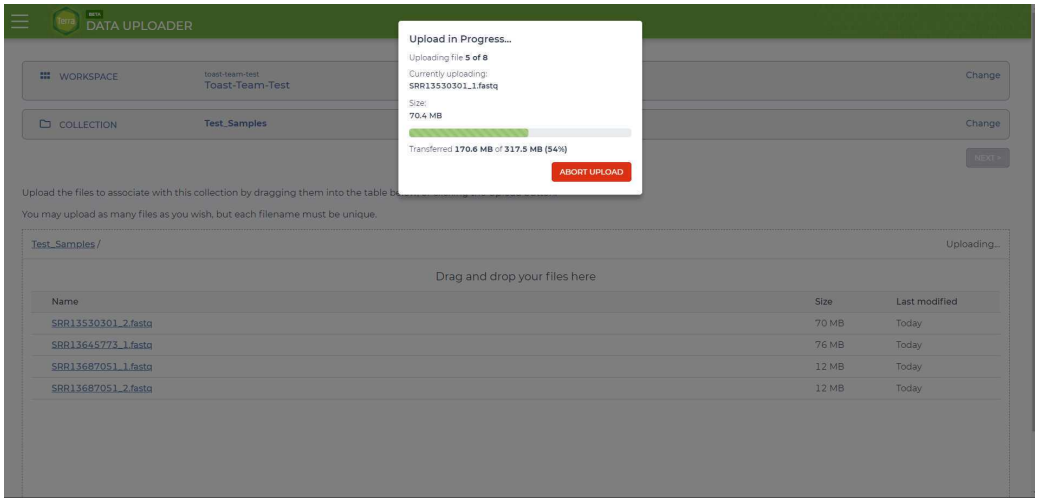
Select the Terra Workspace to which you would like to upload your fastq files. This will be the same workspace created in the previous step.

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.**



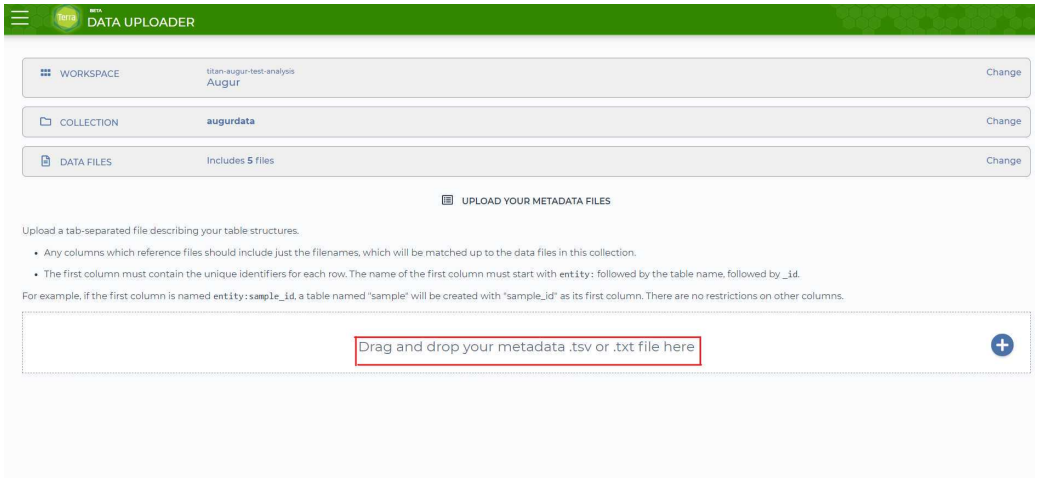
Click the 'Create Collection' button and you will be routed to the data uploader page where you can drag and drop fastq files directly into your browser window to initiate upload.

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

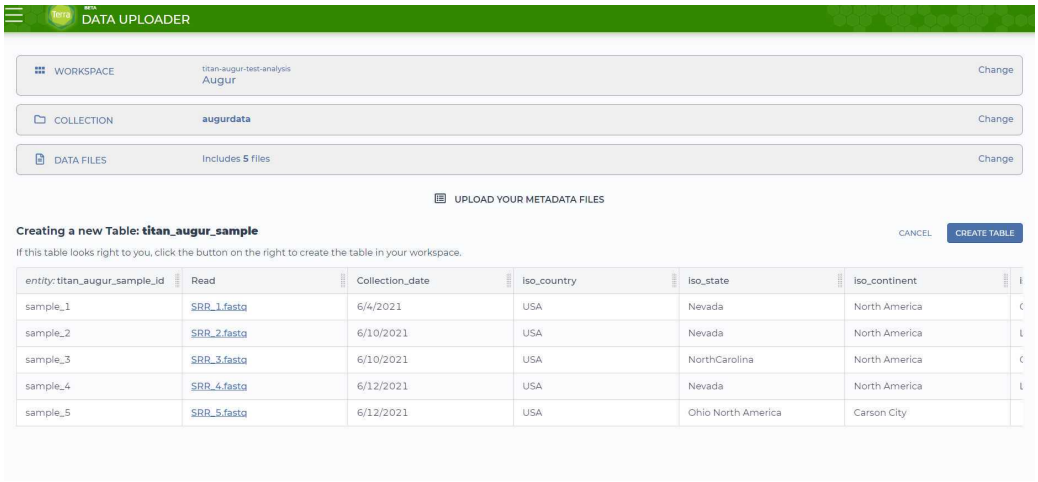


Screen when files are uploading

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



Drag and drop your table data



DATA UPLOADER page after creating new table metadata for the samples

Here is a video of the process:

Running the Augur Workflows

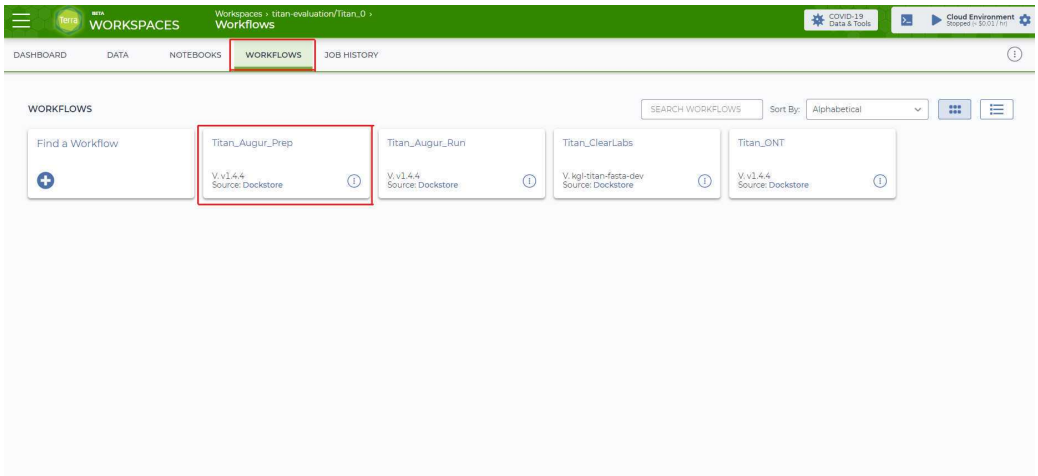
4 Running Titan_Augur is a two step process.

1. Run Titan_Augur_Prep workflow
2. Run Titan_Augur_Run workflow

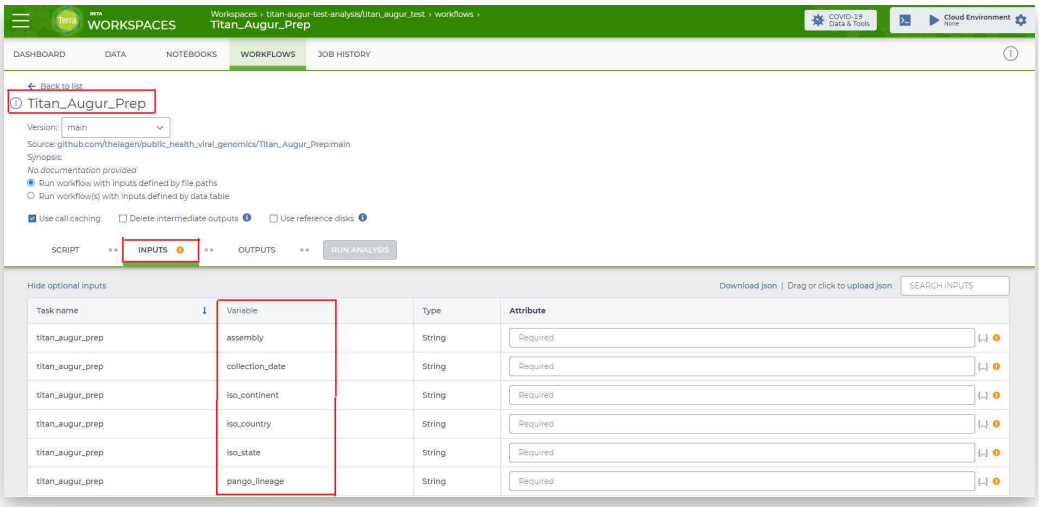
4.1 Running Titan_Augur_Prep workflow

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:



Titan_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 active development and are NOT recommended for users.**

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!**

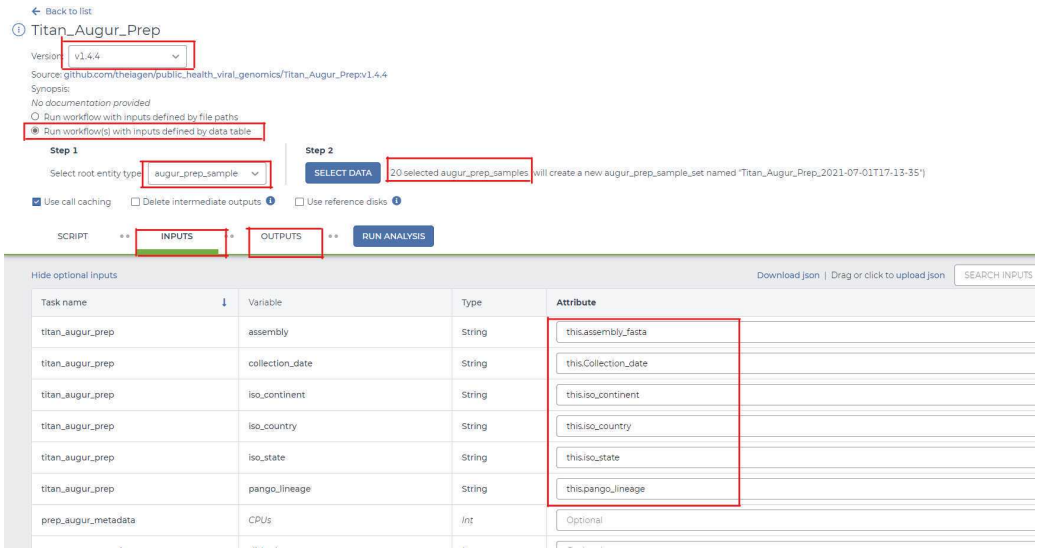
> Call caching allows Terra to identify and skip jobs that have been run previously; this option is by default enabled to avoid unnecessary compute costs. More information on Terra call caching, including examples of when you may want to disable this feature, is available through the Terra Support Documentation.

Click "SELECT DATA" and choose the samples you wish to analyze. **Note that only 25 samples are 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.



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 Titan_Augur_Prep Run:

Once the Titan_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 Titan_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 Titan_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 active development and are NOT recommended for users.**

Ensure that "Run workflow(s) with inputs defined by data table" is selected and click the "Use call caching" box. **In contrast to the Titan_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.

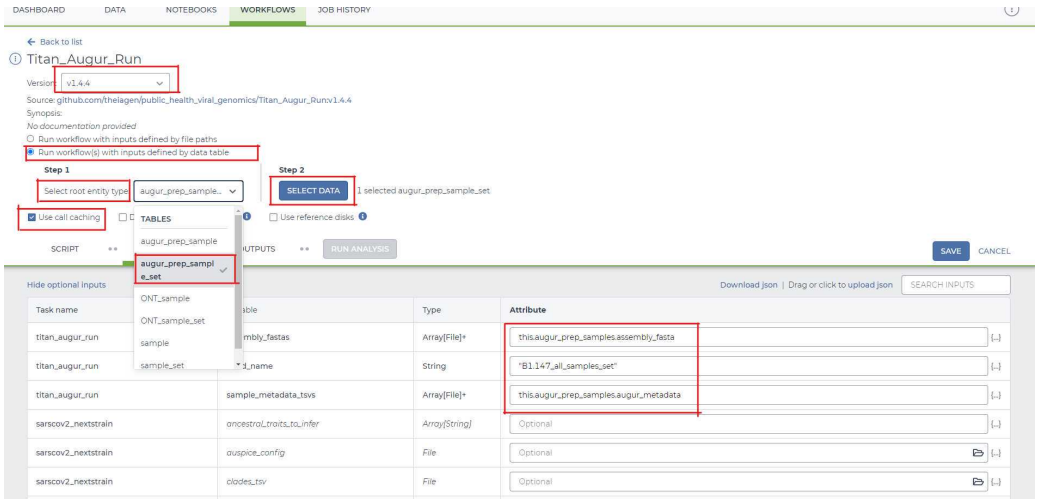
And in INPUTS section, assembly_fastas attribute is set to "this.augur_prep_**samples**.assembly_fastas", Remember it will look for **set of samples** not just sample, so **name the attribute to root entity and set it to plural**.

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.



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 Titan_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.

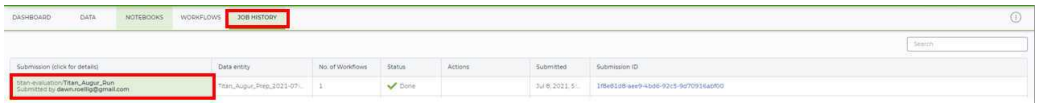
Clicking the 'Launch' button should bring you to the 'Job History' panel where each sample will be queued for the Titan_Augur_Run analysis. The status will change from Queued -> Submitted -> Running

Video showing how to submit the Titan_Augur_Run workflow.

View and Download the Augur workflow output reports

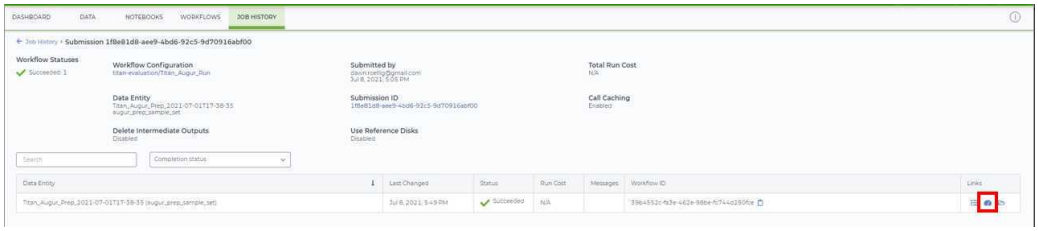
5 First, **verify the sample set has successfully completed** by looking at the 'Workflow Status' section in the top left of the 'Job History' panel. The job has completed when the sample set have a status of 'succeeded' with a green checkmark.

Since Titan_Augur_Run analyzes a set of samples together, the job status would appear this way, it completed analysis successfully (hence the green checkmark).

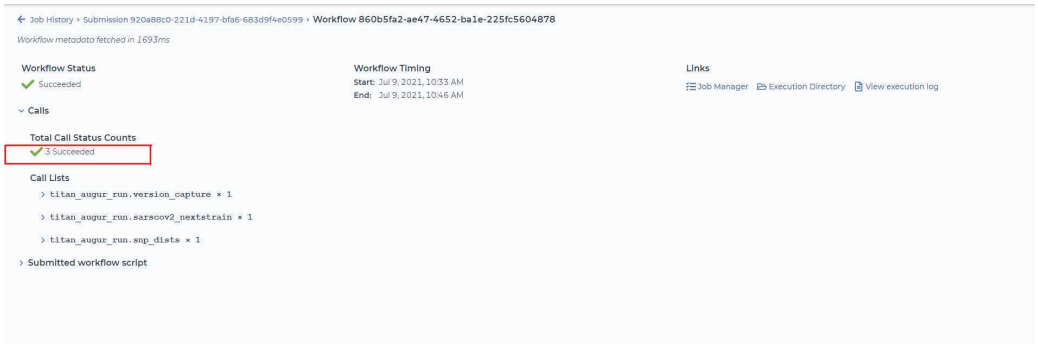


Job Status

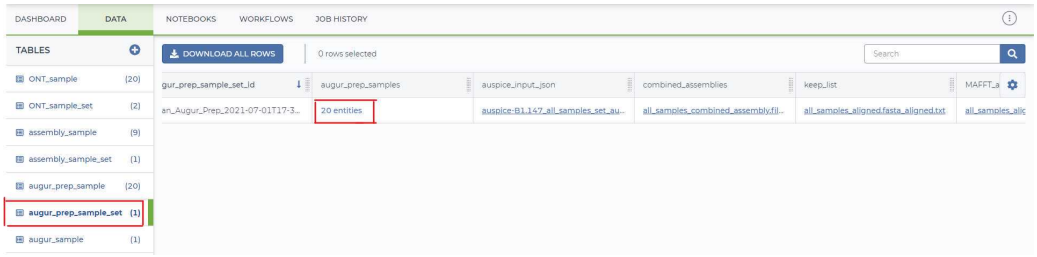
If you want to check what steps have finished within a job (helpful for troubleshooting), then click the job description under the submission column. This will take you to a page with more details on the job run.



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).

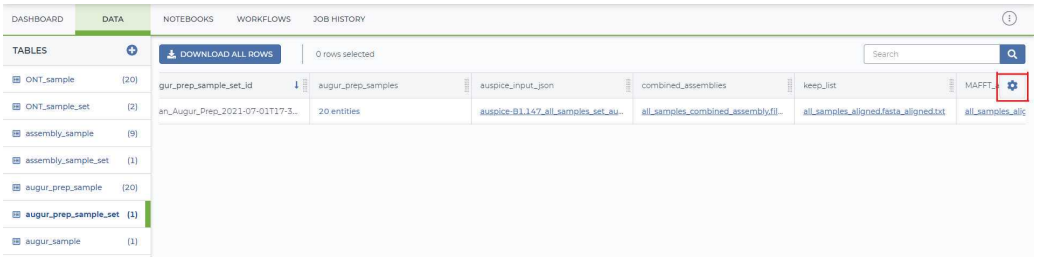


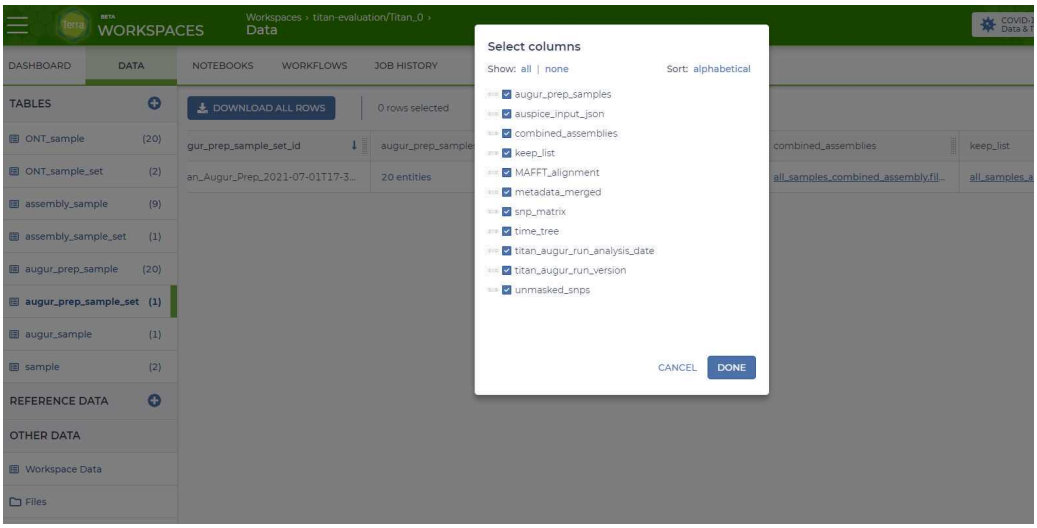
Once confirming the job completed, go to the 'DATA' panel and under the 'TABLES' tab click on the sample table that you created and uploaded in step 4. It will be named 'sample (#)' which has the set of samples/entities in your file



Titan_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".





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

The three files that are used for visualization are:

auspice_input_json - The output generated from the NextClade analysis step 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

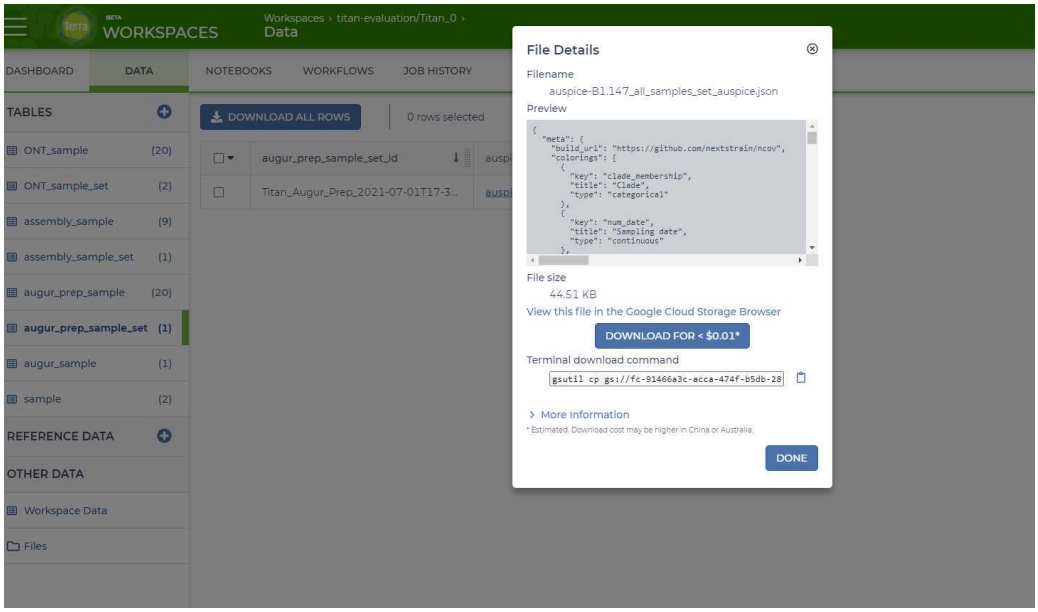
combined_assemblies - A concatenation of all of the assemblies that were included in this phylogenetic analysis. For this example it includes all of 20 samples combined into one single filtered fasta file combined_assemblies.

metadata_merged - Every metadata file from each sample that is generated by Titan_Augur_Prep is merged into one single metadata file.

The specific filenames produced 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 three primary output files of interest

Click the 'Download For < $0.XX' button to save specific files to your local directory for further analyses or visualization.



Download all 3 files now to perform the phylogenetic visualization in the next step.

Video showing how to download and save the output files.

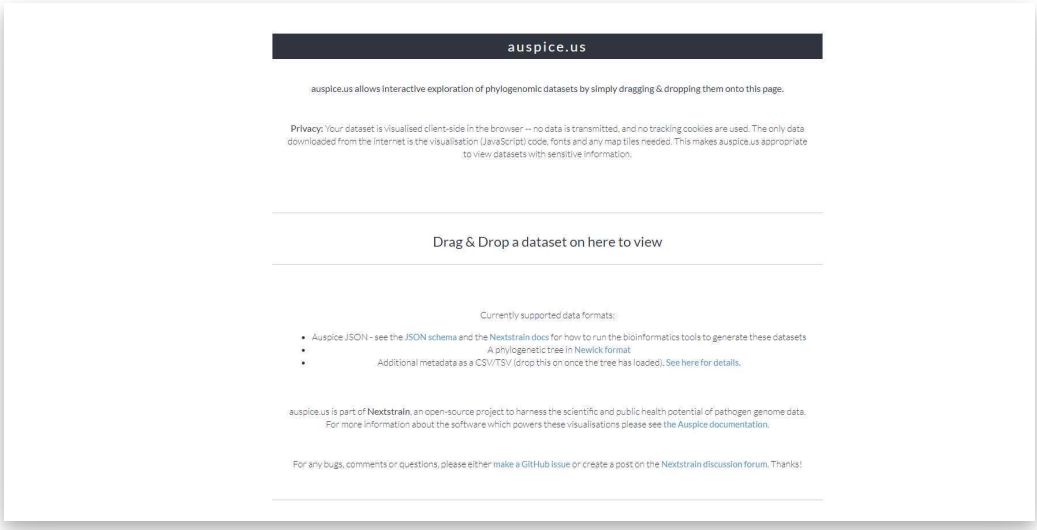
Auspice Visualization

6 After completing the Augur workflow, the resulting output files contain the comparative metrics necessary to analyze the genetic relatedness between samples and draw phylogenetic inference by visualization with Auspice.

Auspice takes the json file to build an interactive phylogenetic tree with Nextstrain. Visit the CDC COVID-19 Genomic Epidemiology Toolkit for an introduction to these applications:

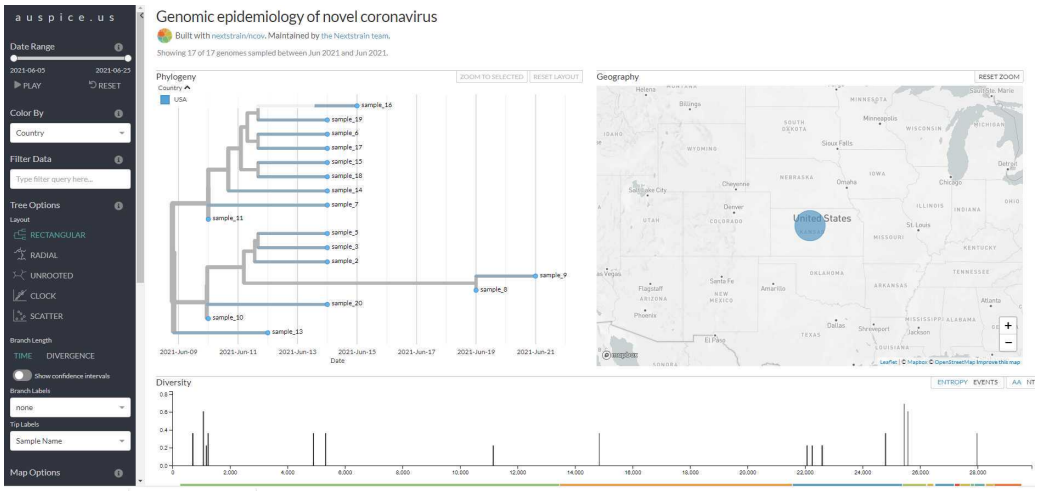
Module 3.2: Getting started with Nextstrain
Module 3.4: Walking through Nextstrain trees

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.



Auspice.us home page

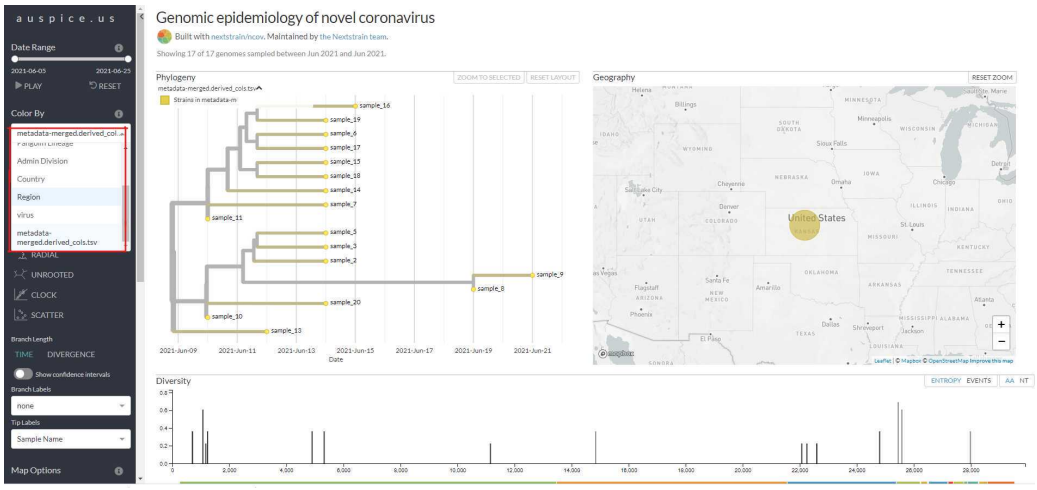
Data can be uploaded to Auspice by simply drag-and-drop of the auspice_input_json file, which should automatically visualize a phylogenetic tree similar the example below:



Example Auspice visualization

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

Metadata can be added to annotate the tree for assistance with identify local outbreaks or clustered samples by simply drag-and-drop of the metadata_merged file onto the 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.

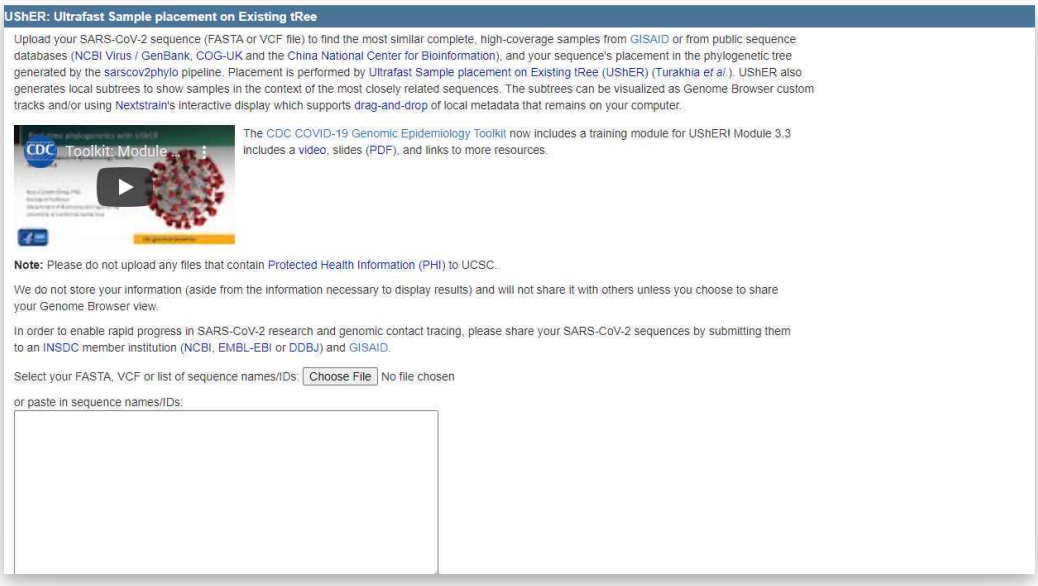
UShER Visualization

7 It may also be helpful to visualize your results using UCSC's UShER webportal.

Visit the CDC COVID-19 Genomic Epidemiology Toolkit for an introduction to UShER:

Module 3.3: Real-time phylogenetics with UShER

Start by navigating to the UShER SARS-CoV-2 webportal: https://genome.ucsc.edu/cgi-bin/hgPhyloPlace



UShER SARS-CoV-2 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 also open the combined_assembly file and then copy and paste the sequence names/IDs.

UShER is very fast and will place your sequences within a phylogenetic tree of public sequences in GISAID or GenBank within 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 subtree results with the 50 sequences (default) most-closely related 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.

Subtrees generated by UShER depend on the divergence among the input samples, if the strains are more divergent, it will produce more subtrees, and if you wish to add more samples, UShER can add them to the existing tree based on sample divergence and mutation. Trees can then be visualize and explore this using the Nextstrain platform.

In addition to phylogenetic placement, UShER also provides extra information, including QC metrics, in table format on the UShER results page.

If you wish to view the subtrees, select the options that is available below the table to launch the phylogenetic tree on the Nextstrain platform. Metadata can be added by drag-and-drop similar to with Auspice (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 Newick files |

Fasta Sequence	Size (?)	#Ns (?)	#Mixed (?)	Bases aligned (?)	Inserted bases (?)	Deleted bases (?)	#SNVs used for placement (?)	#Masked SNVs (?)	Nextstrain clade (?)	Neighboring sample in tree (?)	Lineage of neighbor (?)	#Imputed values for mixed bases (?)	#Maximally parsimonious placements (?)	Parsimony score (?)	Subtree number (?
sample_10	29782 (?)	514 (?)	0	29268 (?)	0	0	4 (?)	0	20C	Wales/PHWC-2960F/2020 | 2020-04-03	?	0	56	0	2 (view in Nextstrain)
sample_11	29782 (?)	0 (?)	0	29782 (?)	0	0	6 (?)	0	20A	IHUCOVID-0483 | LR794668.1 | ?	B.1.147	0	1	0	3 (view in Nextstrain)
sample_13	29773 (?)	0 (?)	0	29773 (?)	0	9 (?)	9 (?)	0	20A	USA/MI-MDHHS-SC20148/2020 | MT459273.1 | 2020-03-24	B.1	0	1	3	4 (view in Nextstrain)
sample_14	29782 (?)	0 (?)	0	29782 (?)	0	0	8 (?)	0	20A	IHUCOVID-0483 | LR794668.1 | ?	B.1.147	0	1	2	5 (view in Nextstrain)
sample_15	29782 (?)	1666 (?)	0	27916 (?)	bases 55 - 29836 align to reference bases 55 - 29836		6 (?)	0	20A	IHUCOVID-0483 | LR794668.1 | ?	B.1.147	0	2	0	6 (view in Nextstrain)
sample_16	29782 (?)	0 (?)	0	29782 (?)	0	0	6 (?)	0	20A	IHUCOVID-0483 | LR794668.1 | ?	B.1.147	0	1	0	7 (view in Nextstrain)
sample_17	29782 (?)	0 (?)	0	29782 (?)	0	0	6 (?)	0	20A	IHUCOVID-0483 | LR794668.1 | ?	B.1.147	0	1	0	8 (view in Nextstrain)
sample_18	29494 (?)	1068 (?)	0	28405 (?)	0	0	5 (?)	0	20A	IHUCOVID-0483 | LR794668.1 | ?	B.1.147	0	3	0	9 (view in Nextstrain)
sample_19	29782 (?)	526 (?)	0	29256 (?)	0	0	6 (?)	0	20A	IHUCOVID-0483 | LR794668.1 | ?	B.1.147	0	2	0	10 (view in Nextstrain)
sample_2	29773 (?)	0 (?)	0	29773 (?)	0	9 (?)	7 (?)	0	20C	USA/UT-UPHL-201121082/2020 | MW210972.1 | 2020-06-19	B.1.446	0	26	1	11 (view in Nextstrain)
sample_20	29782 (?)	2356 (?)	0	27426 (?)	0	0	5 (?)	0	20A	England/LIVE-9E869/2020 | 2020-04-11	?	0	507	1	12 (view in Nextstrain)
sample_3	29782 (?)	2144 (?)	0	27638 (?)	0	0	6 (?)	0	20C	USA/UT-UPHL-201121082/2020 | MW210972.1 | 2020-06-19	B.1.446	0	235	0	13 (view in Nextstrain)
sample_5	29782 (?)	2144 (?)	0	27638 (?)	0	0	6 (?)	0	20C	USA/UT-UPHL-201121082/2020 | MW210972.1 | 2020-06-19	B.1.446	0	235	0	14 (view in Nextstrain)
sample_6	29782 (?)	0 (?)	0	29782 (?)	0	0	6 (?)	0	20A	IHUCOVID-0483 | LR794668.1 | ?	B.1.147	0	1	0	15 (view in Nextstrain)
sample_7	29782	0 (?)	0	29782 (?)	0	0	9 (?)	0	20A	IHUCOVID-0483 | LR794668.1 | ?	B.1.147	0	1	3	16 (view in

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.

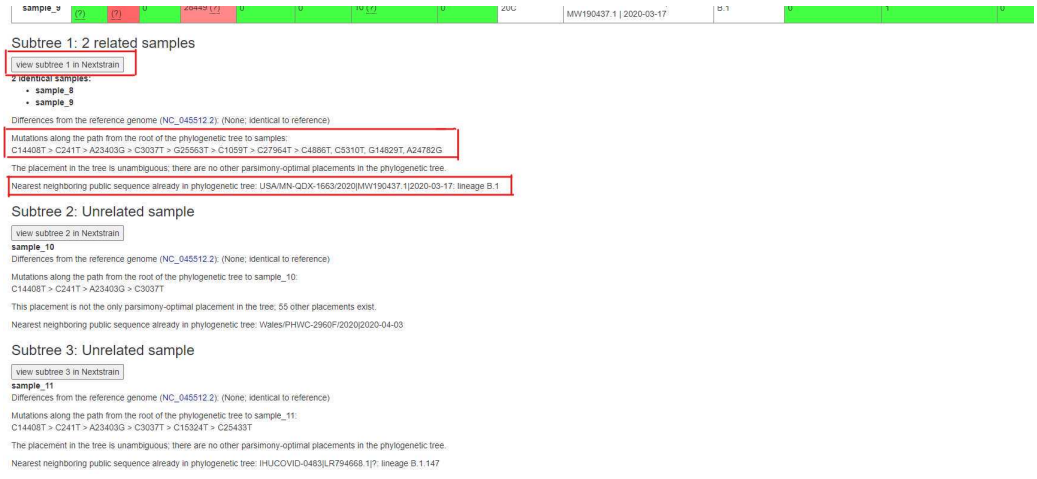
If you want to see the subtrees click the subtree section of the row you are interested in and the link associated to that sample will redirect to the subtree visual rendered on the Nextstrain platform. Nextstrain distinguishes the public samples in Black and your input samples in Red. Subtrees basically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

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 subtrees by scrolling down the page and selecting from the individual subtree.

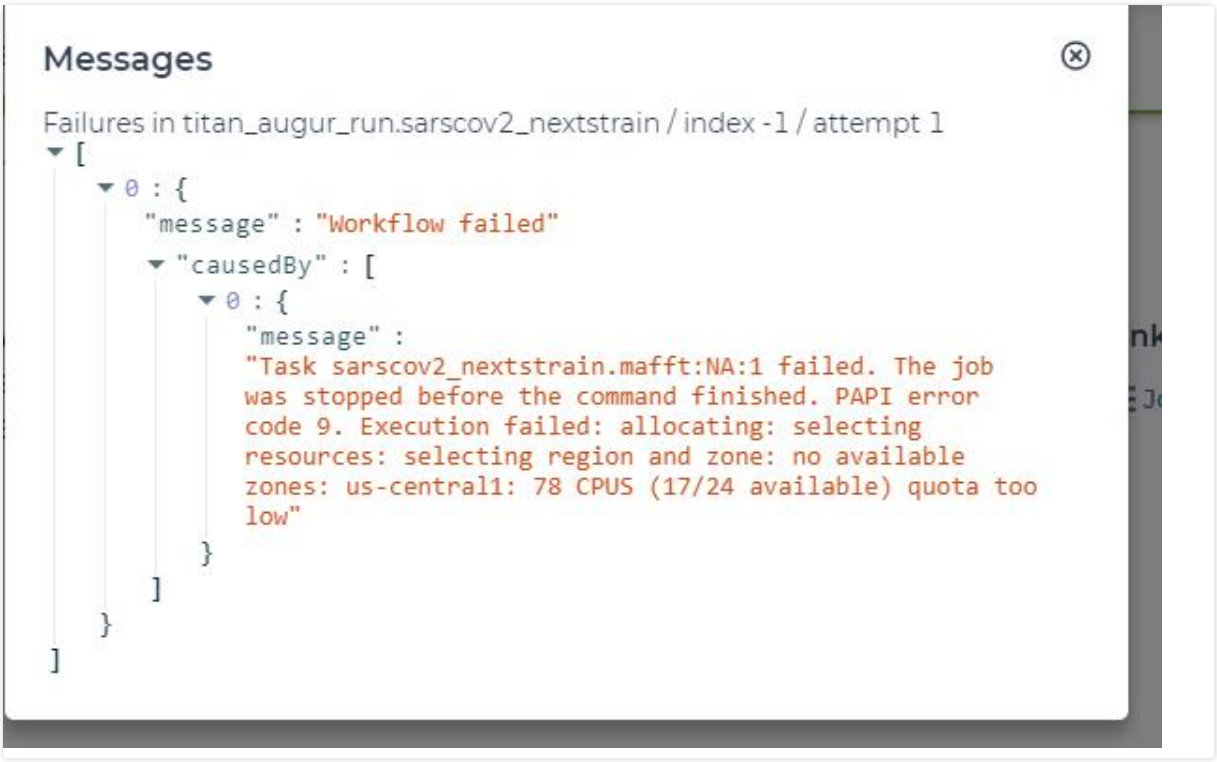
Additional information, such as the likely path of the mutation from root of the tree to your sample, most closely-related sequence in the global tree, and PANGO lineage inference, are provided.



How to debug a failed run

8

1. Failure due to the unavailability of the compute environment.
The Titan_Augur_Run workflow script requests 78 CPUs in order to perform the phylogenetic reconstruction for visualization in Auspice and Nextstrain. If the analysis fails due to lack of resources, try requesting for more CPUs to Terra.



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 Titan_Augur_Run:
- Singular instead of plural notation
- Using sample instead of Sample_Set

First, the singular instead of plural problem:

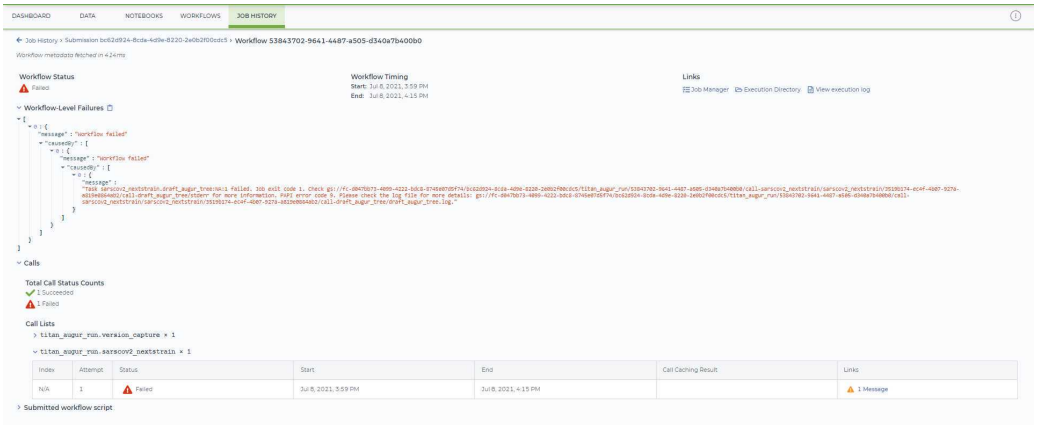
The Titan_Augur_run workflow looks for concatenated set of fasta files, **you need to set the root entity to "name_sample_set" (with name being specific to your dataset) and the input attribute notation of "this.name_samples.assembly_fasta and metadata as "this.name_samples.metadata"** rather than using "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:

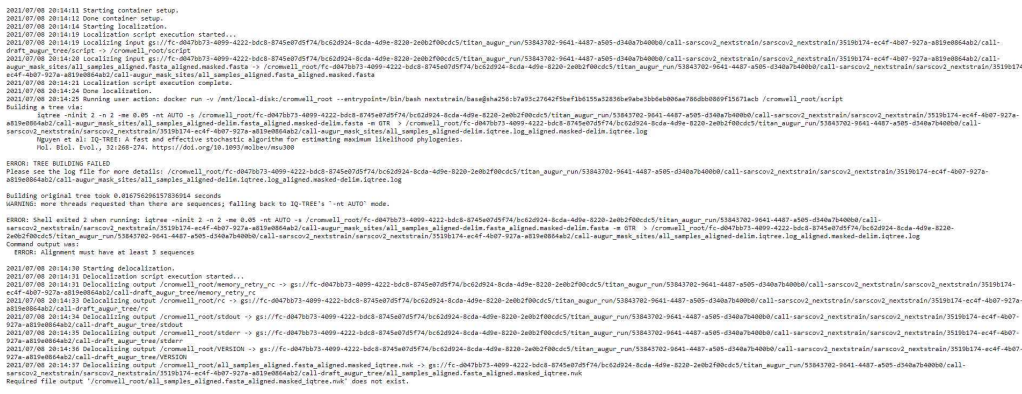
Example error caused by singular instead of plural in inputs

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



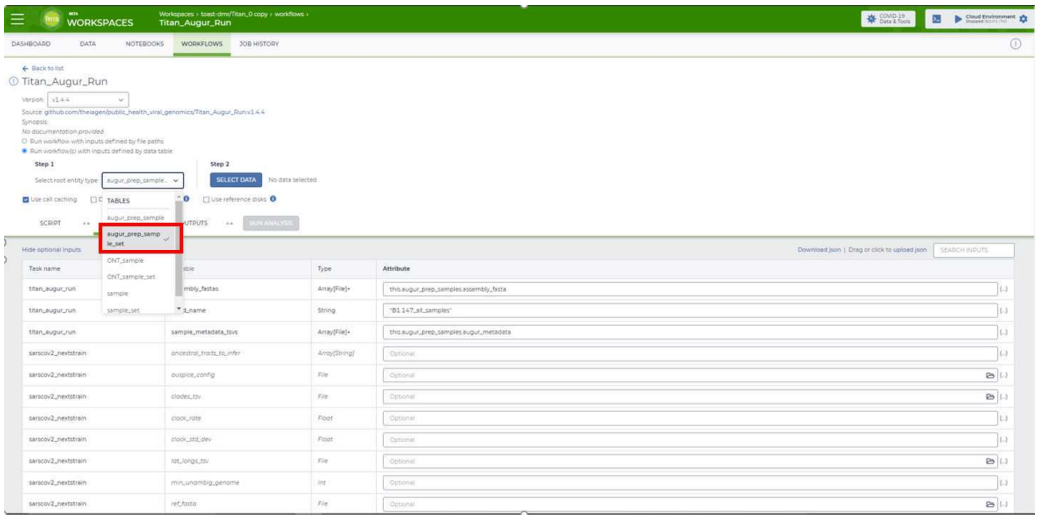
Error in Terra.bio job history

The specific error message can be found in the draft_augur_tree.log file in the specified Google Cloud storage location.



Example draft_augur_tree.log output with error message

To avoid this error, ensure the root entity type is set to 'sample_set' and not 'sample', as shown below.

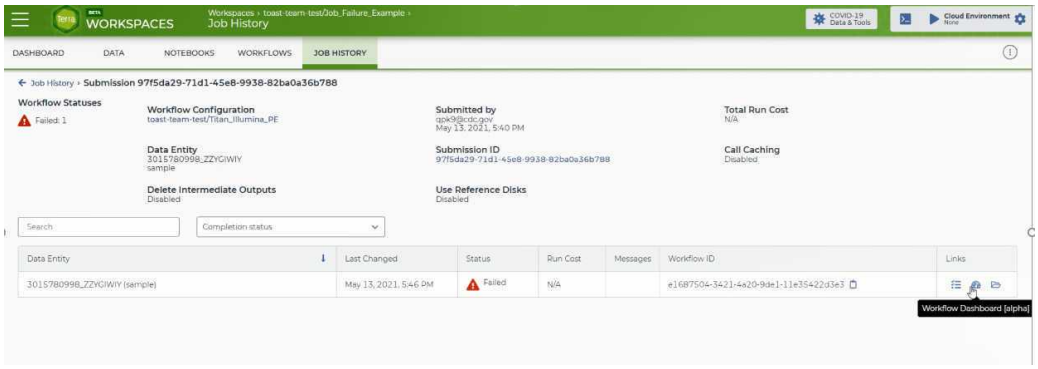


Make sure the root entity type is 'sample_set'

3. Error due to spacing in collection name

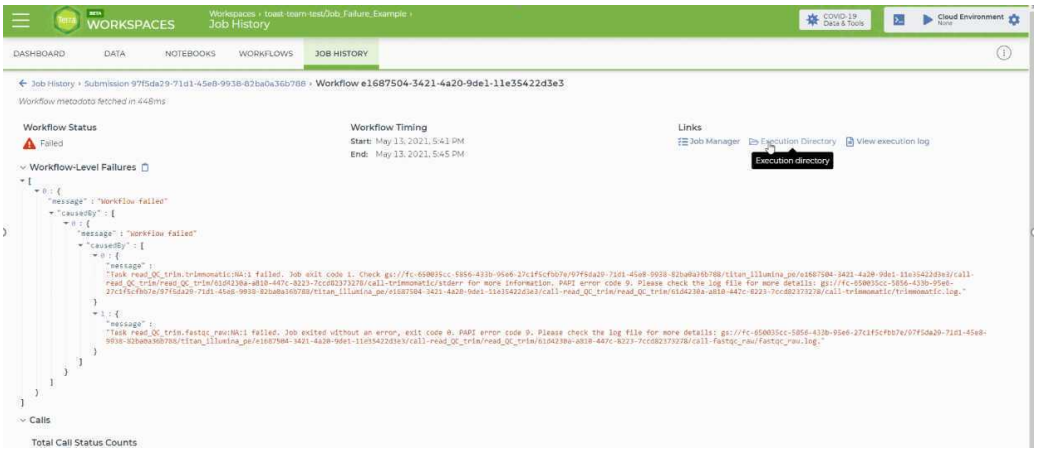
In this example of a failed run we have an error caused by adding spaces into your collection name. If there is a failure for some other reason you can follow similar steps to understand why your job failed. Here we will be using Illumina paired-end data, but the process of debugging an error is the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

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



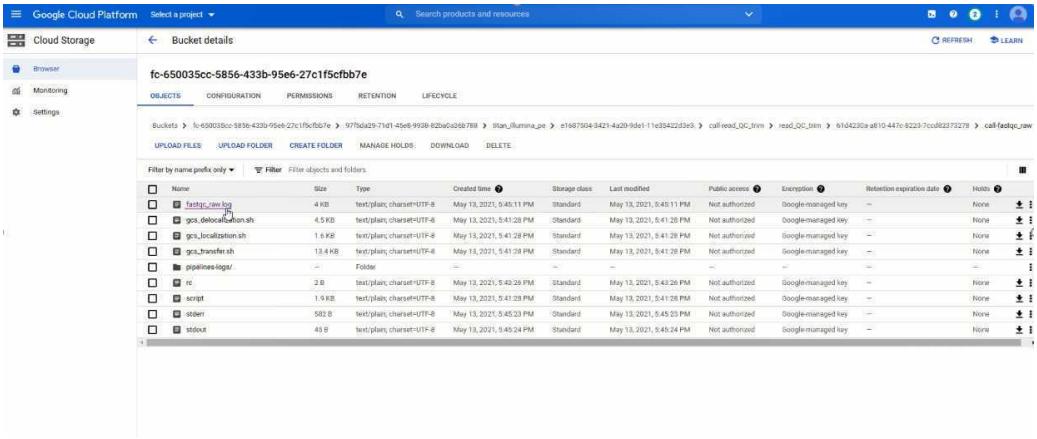
Job Failure

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.



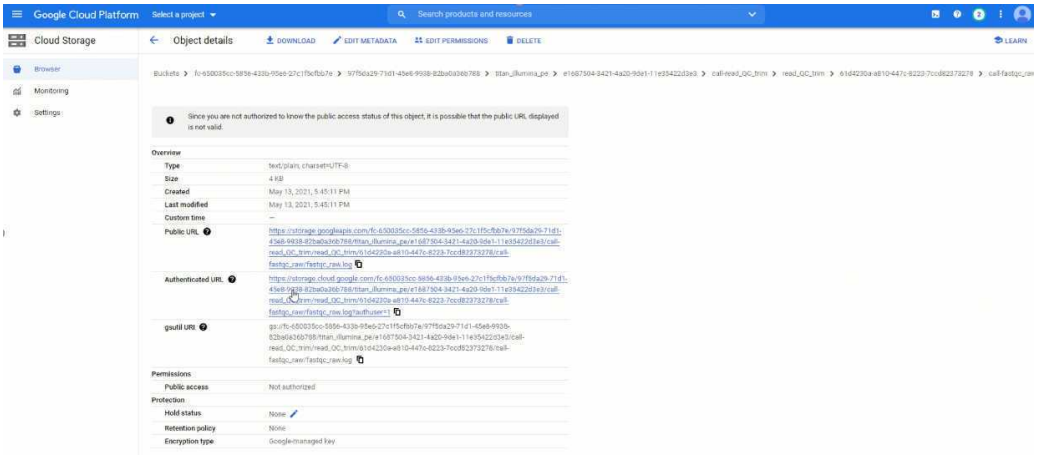
Job failure messages

Follow the file path to the Google Cloud storage bucket location to the log files referenced in the error messages (in red/orange text in the above photo).



Click on log file referenced in the error message.

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



Authenticated URL link

In the text file we can see that there was an error that cause 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.

Example fastqc_raw.log file showing the error.

Video of the whole process.

Some error message can be difficult to understand. Contact **TOAST@cdc.gov** for help debugging job failures.