

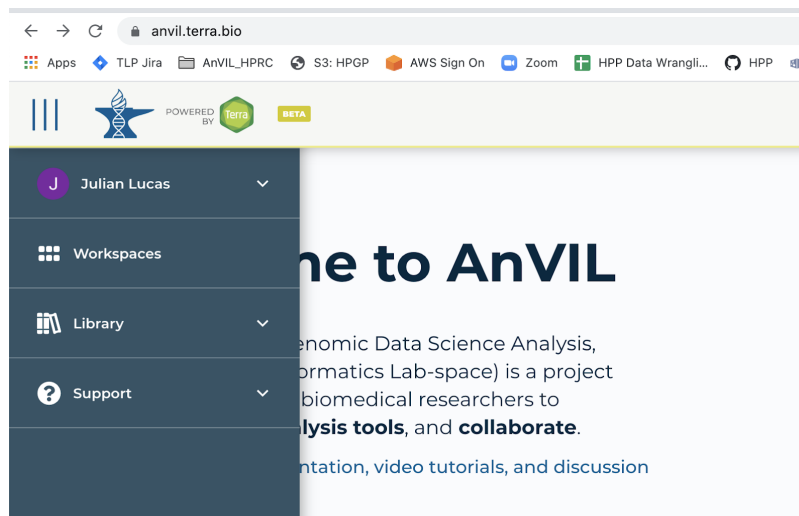
Instructions for Giraffe/DeepVariant Demonstration Workspace

The following instructions are for the [HPRC-Giraffe-Demo-2023](#) workspace in AnVIL. These instructions and the workspace they refer to were created for HPRC and AnVIL workshops. While following these instructions, users will learn how to clone a workspace, run a workflow, and inspect the results of the workflow in a Jupyter Notebook. Running this exercise should cost less than approximately \$1 dollar.

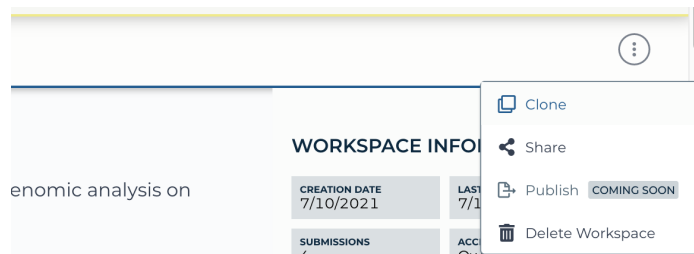
Prerequisites: Create a Terra account following [this guide](#).

I. Clone The Pre-Prepared Workspace

1. Install Chrome (if you don't already have it installed)
 - a. AnVIL (currently) requires Chrome, so this step is not optional
2. Open AnVIL
 - a. Open your Chrome browser and navigate to <https://anvil.terra.bio/>
 - b. Login if necessary (you will need to use your gmail account that you used to sign up for Terra)
 - c. If you have trouble logging in please contact one of the instructors!
3. Open the pre-prepared Giraffe/Deepvariant workspace
 - a. Click on the hamburger icon



- b. Select **Workspaces**
 - c. Enter “demo-2023” into the search bar (top center of screen)
 - d. Select the **PUBLIC** tab
 - e. Select the **HPRC-Giraffe-Demo-2023** workspace
4. Clone the workspace
 - a. Click the three vertical dot icon on the upper right-hand corner of the screen.



- b. Select **Clone**
- c. Fill out the fields of the popup window

**Note: Choose a unique name for your cloned workspace. You can just enter your initials at the end of the existing workspace name, if you like.*

**Note: If you have your own billing project it is ok to use that (the analysis in this workspace cost less than \$1 to run – because the data has been downsampled).*


- d. Click **CLONE WORKSPACE**


Check in: at this point, you have copied the workspace and you have your own version! This means that you can run analysis and the results will be your own. You can also add in your own data or analysis steps if you want.

II. Run The Giraffe/DeepVariant Workflow

1. Spend a few minutes familiarizing yourself with the workspace.
 - a. Read through the **DASHBOARD**
 - b. Click on the **DATA** tab and look at the sample table. All of the columns with “input_” prepended to the name will be used to run our workflow.
2. Click on the **WORKFLOWS** tab and select the **GiraffeDeepVariantLite** workflow
 - a. Your screen should look very similar to the screenshot below:

[← Back to list](#)

 GiraffeDeepVariantLite

Version: giraffe-dv-dt-hprc... 

Source: github.com/vgteam/vg_wdl/GiraffeDeepVariantLite:giraffe-dv-dt-hprcy1


Synopsis:

No documentation provided

☐ Run workflow with inputs defined by file paths




☒ Run workflow(s) with inputs defined by data table

Step 1

Select root entity type: sample 

Step 2

SELECT DATA No data selected

☒ Use call caching ☐ Delete intermediate outputs  ☐ Use reference disks  ☐ Retry with more memory 

3. Select samples to run
 - a. Click **Run workflow with inputs defined by data table** if it is not already selected
 - b. If not done already, **select root entity type: sample**. This means that you are referencing the sample data table in your workspace.
 - c. Click **SELECT DATA**
 - d. A popup will open, click the check box for the first row, then press **OK** (bottom right-hand corner of the popup)
4. Familiarize yourself with the input parameters
 - a. Click on the **INPUTS** tab (should be on the center or bottom of the page)
 - b. This tab has been pre-populated for you, but look through the inputs. You should see:
 - i. Pangenome data structures from the workspace
 1. These start with “workspace.”
 - ii. Inputs from the sample table
 1. These start with “this.” (anything after the dot references a column in the table).
 - iii. Manually entered run parameters
 1. Search for the term “CORES” for examples
 2. These could have been put into the sample table too, if you want to run a sample multiple times with different conditions!
5. Familiarize yourself with the outputs parameters
 - a. Click on the **OUTPUTS** tab
 - b. This too has been pre-populated
 - c. Any entries that start with “this.” will write to a column in the sample data table (this is really handy for keeping track of your results)
6. You are ready to start, click **RUN ANALYSIS** button
 - a. You will see a popup, click the **LAUNCH** button
7. Monitor your job
 - a. After launching, you are redirected to the JOB HISTORY tab.

- b. Keep this open
- c. If you see a status with a red triangle (below), your workflow has failed.



- i. Workflows will often fail after about 2-3 minutes. If that happens to you, your input and output parameters are likely incorrect. Repeat the steps above and try to launch again. If you have repeated failures, reach out to an instructor for assistance.
- d. If you see a status with a green check mark, your workflow is done!

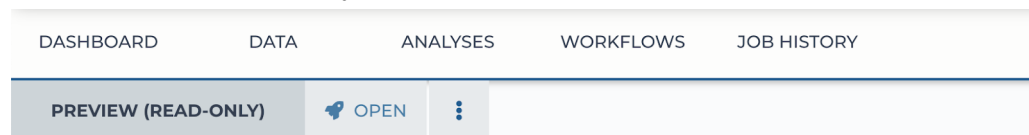


Check in: congratulations, you just ran a workflow in AnVIL! It should be -- and often is -- this easy. If you want to investigate something with a tool that someone else wrote, you don't have to install software, or rewrite the tool yourself.

III. Inspect Workflow Results With A Jupyter Notebook

In this section we are going to launch a Jupyter notebook to show how Terra allows users to view and analyze data held in a workspace.

- 1. Launch the example notebook
 - a. Click the **ANALYSES** tab at the top of your screen
 - b. Click on the Jupyter notebook (not the prerun one)
 - i. What you see at first is a preview, the notebook is not running yet
 - c. Click **OPEN** on the top of your screen



Environment

- d. Create your cloud environment
 - i. You will see a popup on the right-hand side of your screen
 - ii. At the bottom, next to **Create custom environment**, click **CUSTOMIZE**
 - iii. Click the dropdown, and select Custom Environment

Application configuration ⓘ

Default: (GATK 4.2.4.0, Python 3.7.12, R 4.2.1) ▼

Default: (GATK 4.2.4.0, Python 3.7.12, R 4.2.1) ✓

Legacy GATK: (GATK 4.2.4.0, Python 3.7.12, R 4.1.3)

Legacy R / Bioconductor (R 4.1.1, Bioconductor 3.13, Python 3.7.10)

COMMUNITY-MAINTAINED JUPYTER ENVIRONMENTS (VERIFIED PARTNERS)

Pegasus (Pegasuspy 1.6.0, Python 3.7.12, harmony-pytorch 0.1.7, nmf-torch 0.1.1, scVI-tools 0.16.0)

OTHER ENVIRONMENTS

Custom Environment

- iv. Select 1 CPUs
- v. Enter `jmonlong/terra-notebook-vg:1.1` into the **Container image** box

Application configuration ⓘ

Custom Environment ▼

Container image

`jmonlong/terra-notebook-vg:1.1`

Custom environments **must** be based off one of the [Terra Jupyter Notebook base images](#)

Startup script *Optional*

URI

Creation Timeout Limit ⓘ

10 Minutes

Cloud compute profile

CPUs `1` Memory (GB) `3.75`

☐ Enable GPUs **BETA** [Learn more about GPU cost and restrictions.](#)

- vi. Scroll down to press **Next**, then **CREATE**
 1. It is ok that the Docker image is unverified
 - vii. It will take a few (3-5) minutes to create your environment. (Terra is launching a virtual machine to run your notebook.)
 - viii. Once the environment is ready, you can click **OPEN**
2. Run the notebook
 - a. Follow the instructions in the notebook
 - b. If you aren't familiar with Jupyter notebooks, you can execute a command by clicking on a cell and pressing **Shift + Return**
 - c. Once you are done running the notebook, there is no need to shut the cloud environment down. Terra will automatically shut down environments that have been idle for a period of time

Check in: we have now run a workflow and analyzed the results by:

- *Cloned a workspace*
- *Run a workflow using data organized in a data table*
- *Taken advantage of data stored in other workspaces (HPRC_AnVIL, for example)*
- *Inspected the results of the workflow, and (optionally) uploaded a modified VCF to a new data table*

By running workflows and creating new data tables, you can chain workflows together, analyze your runs, and document the process all without having to move data from the cloud to your local machine. Pretty cool.