GDSCN Book: SARS with Galaxy on AnVIL

November, 2021

Table of Contents

# About this Book

This book provides resources for instructors to engage students in a cloud-based Galaxy activity on AnVIL, focused on SARS-CoV-2 variant detection.

# 1 Overview

# 2 Learning Objectives

# 3 Lesson Plan

# 4 What Is a Variant?

# 5 The Sequencing Revolution

# 6 Alignments

# 7 Cloud Computing

# 8 Overview Video

# 9 Student Activity Guide

This chapter contains the student instructions for the SARS-CoV-2 Variant Detection with Galaxy activity.

## 9.1 Introduction

Galaxy is a free, relatively easy to use bioinformatics implementation package. It changes command line programs into GUI based programs.

### 9.1.1 Before You Start

Make sure you have an AnVIL account (refer to <https://jhudatascience.org/AnVIL_Book_Getting_Started/student-account-setup.html>)

### 9.1.2 Objectives

This activity will teach you how to use the AnVIL platform to:

1. Get started working on AnVIL
2. Launch the Galaxy tool
3. Examine fastq sequence data files
4. Align sequence data to a reference genome
5. View the aligned data and reference genomes interactively

## 9.2 Getting Started

### 9.2.1 Set Up

In the next few steps, you will walk through how to get set up to use Galaxy on the AnVIL platform. AnVIL is centered around different “Workspaces”. Each Workspace functions almost like a mini code laboratory - it is a place where data can be examined, stored, and analyzed. The first thing we want to do is to copy or “clone” a Workspace to create a space for you to experiment.

Use a web browser to go to the AnVIL website. In the browser type:

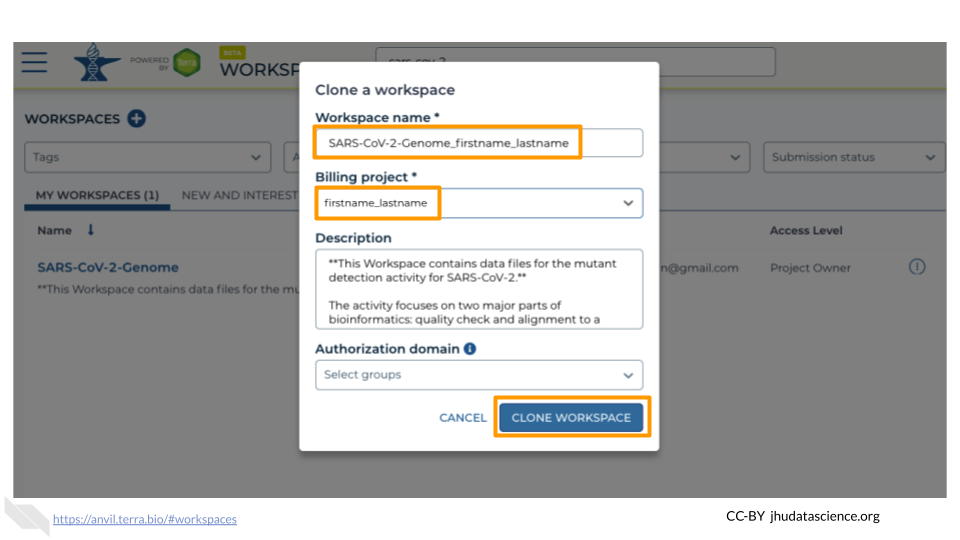
anvil.terra.bio

After logging in, click “View Workspaces”. Select the “Public” tab. In the top search bar type the activity workspace “SARS-CoV-2-Genome”. You can also go directly to the following link: <https://anvil.terra.bio/#workspaces/gdscn-exercises/SARS-CoV-2-Genome>.

Clone the workspace by clicking the teardrop button (). And selecting “Clone”.



In the first box, give your Workspace clone a name by adding an underscore (“\_”) and your name. For example, “SARS-CoV-2-Genome\_Ava\_Hoffman”. Next, select the Billing project provided by your instructor. Leave the bottom two boxes as-is and click “CLONE WORKSPACE”.



### 9.2.2 Starting Galaxy

Galaxy is a great tool for performing bioinformatics analysis without having to update software or worry too much about coding. In order to use Galaxy, we need to create a cloud environment. This is like quickly renting a few computers from Google as the engine to power our Galaxy analysis.

In your new Workspace, click on the “NOTEBOOKS” tab. Next, click on “Create a Cloud Environment for Galaxy”. You should see a popup window on the right side of the screen. Click on “NEXT” and “CREATE” to keep all settings as-is. This will take 8-10 minutes. When it is done, click “LAUNCH GALAXY”.



You can also follow along with the first ~2 minutes of [this video](https://jhudatascience.org/AnVIL_Book_Getting_Started/starting-galaxy.html) to start Galaxy on AnVIL.

### 9.2.3 Navigating Galaxy

Notice the three main sections.

**Tools** - These are all of the bioinformatics tool packages available for you to use.

**The Main Dashboard** - This contains flash messages and posts when you first open Galaxy, but when we are using data this is the main interface area.

**History** - When you start a project you will be able to see all of the documents in the project in the history. Now be aware, this can become very busy. Also the naming that Galaxy uses is not very intuitive, so you must make sure that you label your files with something that makes sense to you.



On the welcome page, there are links to tutorials. You may try these out on your own. If you want to try a new analysis this is a good place to start.

### 9.2.4 Importing Data into Galaxy

Luckily, we linked to the original data when we cloned our Workspace! We have three files we will need for our activity. These are (1) the reference genome for SARS-CoV-2, and both forward (2) and reverse (3) reads for our sample. There are two sets of reads for our sample because the scientists who collected it used paired-end sequencing. The reference genome ends in “.fasta” because it has already been cleaned up by scientists. The sample we are looking at ends in fastq because it is raw data from the sequencer.

1. Click on “Upload Data” in the Tools pane.

* 

1. Click on “Choose remote files” at the bottom of the popup. Double-click the workspace folder, then “Tables/” then “reference/”. Click the reference .fasta file so that it is highlighted in green and click “OK”.

* 

1. Now that your reference has been added, click “Choose remote files” again to add the two sample files. Double-click the workspace folder, then “Tables/” then “samples/”. Click the two sample fastq files so that they are highlighted in green and click “OK”.

* 

1. Click “Start” and once complete, you can click “Close”.
2. Confirm your upload worked by looking at the file names in the History pane.

* 

## 9.3 Exercise One: Examining fastq Files in Galaxy

*Now we have some data in our account we can look at it. In this exercise we will see data in fastq format. This is the typical output from an Illumina Sequencer, but also the standard format for most alignment software.*

### 9.3.1 Examining Inputs

Use your mouse and click on the eye icon () of the first fastq file (**VA\_sample\_forward\_reads.fastq**). In the Main screen you will see something like this:



QUESTIONS:

1. How many lines in a .fastq file represent an individual read?
2. What does each line represent?
3. Why is the final line for each read (the quality score) important?

### 9.3.2 Quality Scoring

FastQC is a tool which aims to provide simple quality control checks on raw sequence data coming from high throughput sequencing pipelines. It provides a set of analyses which you can use to get a quick impression of whether your data has any problems of which you should be aware before doing any further analysis.

Find the FastQC tool in the GENOMIC FILE MANIPULATION: FASTQ Quality Control tool folder. You will see something like this in the tools:



In the first drop down menu, make sure it has your first fastq file (**VA\_sample\_forward\_reads.fastq**) loaded. Leave everything else as-is and click on the blue execute button at the bottom of the screen.

The main dash will highlight in green if everything is okay. In the history, you will see the new files turn yellow, then green. If the job fails it will show an error.

Click on the eye icon () in the new file in the history “FASTQC on data2 Webpage”.



You will open up a summary report for the sequencing file:



QUESTIONS:

1. Explore “Basic Statistics”. How many total reads are there? Have any been flagged as poor quality? What is the sequence length?
2. Explore “Per base sequence quality”. Based on the Basic Statistics, is 28-40 a good or bad quality score?
3. Is it okay to proceed based on the per base sequence quality?

**Breakout Box: Learn more about quality scores**

You may be wondering how the fourth line of the .fastq files relates to the quality score above. To save space, the sequencer records an [ASCII character](http://drive5.com/usearch/manual/quality_score.html) to represent scores 0-42. For example 10 corresponds to “+” and 40 corresponds to “I”. FastQC knows how to translate this. This is often called “Phred” scoring.

What does 0-42 represent? These numbers, when plugged into a formula, tell us the probability of an error for that base. This is the formula, where Q is our quality score (0-42) and P is the probability of an error:

Q = -10 log10(P)

Using this formula, we can calculate that a quality score of 40 means only 0.00010 probability of an error!

## 9.4 Exercise Two: Alignment

*Given that our data has passed some quality checks, we will try to align the data to the reference genome. In this case it is simple, a viral genome. A human sequencing project will generate much larger data sets. There are many aligners, but we will start off looking at a simple aligner BWA-MEM. This example uses paired data.*

We will use our two SARs data files, which are ready for alignment. **VA\_sample\_forward\_reads.fastq** **VA\_sample\_reverse\_reads.fastq**

Now go to GENOMICS ANALYSIS: Mapping and select “Map with BWA-MEM”. This program will align your reads to your SARS reference genome. Some of our reads are >100 base pairs so we will use the MEM option.



First, choose your reference. In the first drop down box change it to “Use a genome from history and build index”. Then choose the SARS reference fasta file that you uploaded as the reference.



Under the “Single or Paired-end reads” ensure the “Paired” option is selected. Now choose your forward and reverse fastq files. Leave other options as-is. *You can learn more about what the alignment software BWA-MEM is doing if you scroll down below the execute button*. Click execute.



The output file is a BAM file, which lists where each read aligns to the reference genome and whether there are any differences. You can click the eye button to preview the results, but the results are not easy to interpret visually (much like the fastq files). Instead you will use a genome viewer in the next step.

QUESTIONS:

1. What is alignment software (for example, BWA-MEM) actually doing?
2. Here we are using paired fastq (“paired end”) data. What is an advantage of using paired data?

## 9.5 Exercise Three: Viewing aligned data

We have aligned our data but it is currently a table of where the reads align. This is hard to read, so we will use JBrowse to view the data.

Go to STATISTICS AND VISUALIZATION: Graph/Display Data and select “JBrowse”. Under “Reference genome to display” ensure that “Use a genome from history” is selected. Below this, make sure that the **SARS-CoV-2\_reference\_genome.fasta** file is selected. Click “Execute”.



You should see a new JBrowse item appear in your history. Click the eye icon () to open JBrowse. You will need to click on the magnifying glasses to zoom in, but you should see the A,C,G, and Ts and their corresponding colors that make up the SARS-CoV-2 genome!



This is interesting, but it doesn’t let us compare the genome to the sample we have. We suspect there may be some differences that indicate our sample is the delta variant. Go back to STATISTICS AND VISUALIZATION: Graph/Display Data and select “JBrowse”. Just like before, under “Reference genome to display”, ensure that “Use a genome from history” is selected. Below this, make sure that the **SARS-CoV-2\_reference\_genome.fasta** file is selected.

This time, we’ll add our alignment data from [Exercise Two](#exercise-two). Click the “+ Insert Track Group” button.



Click “+ Insert Annotation Track” to add our alignment data.



You’ll keep everything the same except the following:

* **Track Type**: BAM Pileups
* **Autogenerate SNP Track**: Yes
* **Maximum size of BAM chunks**: Add one more zero: 50000000



These arguments tell JBrowse what kind of data we are using, as well as some memory options. Click “Execute”.

You should see a new JBrowse item appear in your history. Click the eye icon () to open JBrowse. Make sure that all boxes are checked on the left side: “Available Tracks”. Note that the tracks show up in the order that you click on them.



Let’s look at an example mutation in our sample. Type in the reference position “24410” and click “Go”. You should see a bunch of “A”s highlighted in green throughout our sample. The reference sequence (top line) is a “G” but all of the reads are an “A”. This means that our sample is genetically different from the established SARS-CoV-2 reference genome! Researchers often call these single base differences “SNPs” - Single Nucleotide Polymorphisms.



QUESTIONS:

1. How long is the SARS-CoV-2 genome? Hint: zoom out and scroll to the end of the genome.
2. Locate position 23,603. This is the site of an important mutation in the spike protein of the delta variant “P681R”. In this mutation, the amino acid proline is replaced by arginine. Is this mutation present at position 23,603 in our sample? Based on the evidence, do you think this sample is a delta variant?

**Breakout Box: Sequencing errors**

It’s possible to make mistakes in the data preparation before we get to the data analysis. Sometimes this happens when the samples are being prepared in the lab and sometimes this happens because the sequencer makes a mistake. This is one reason why quality scores are helpful. With millions of reads of data, it’s more likely that we see a “SNP” that is actually an accident. Multiple copies of the same areas of our data (“read depth”) help us be sure it’s a real SNP. When we compare across lots of aligned reads of the same area, we can determine the actual sequence by consensus. For example, we can be reasonably confident that the “G” at position 1,203 shown below is just a sequencing or lab mistake.



## 9.6 Wrap-up

Once you are done with the activity, you’ll need to shut down your Galaxy cloud environment. This frees up the cloud resources for others and minimizes computing cost. The following steps will delete your work, so make sure you are completely finished at this point. Otherwise, you will have to repeat your work from the previous steps.

Return to AnVIL, and find the Galaxy logo that shows your cloud environment is running. Click on this logo:



Next, click on “DELETE ENVIRONMENT OPTIONS”:



Finally, select “Delete everything, including persistent disk”. Make sure you are done with the activity and then click “DELETE”.



# Appendix

# 10 Register and Solutions

# 11 Help

# 12 Galaxy Training Network

More available at <https://training.galaxyproject.org>

# About the Authors

These credits are based on our [course contributors table guidelines](https://github.com/jhudsl/DaSL_Course_Template_Bookdown/wiki/How-to-give-credits).

| Credits | Names |
| --- | --- |
| **Pedagogy** |  |
| Lead Content Instructor(s) | [FirstName LastName](link%20to%20personal%20website) |
| Lecturer(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | Delivered the course in some way - video or audio |
| Content Author(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | If any other authors besides lead instructor |
| Content Contributor(s) (include section name/link in parentheses) - make new line if more than one section involved | Wrote less than a chapter |
| Content Editor(s)/Reviewer(s) | Checked your content |
| Content Director(s) | Helped guide the content direction |
| Content Consultants (include chapter name/link in parentheses or word “General”) - make new line if more than one chapter involved | Gave high level advice on content |
| Acknowledgments | Gave small assistance to content but not to the level of consulting |
| **Production** |  |
| Content Publisher(s) | Helped with publishing platform |
| Content Publishing Reviewer(s) | Reviewed overall content and aesthetics on publishing platform |
| **Technical** |  |
| Course Publishing Engineer(s) | Helped with the code for the technical aspects related to the specific course generation |
| Template Publishing Engineers | [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/) |
| Publishing Maintenance Engineer | [Candace Savonen](https://www.cansavvy.com/) |
| Technical Publishing Stylists | [Carrie Wright](https://carriewright11.github.io/), [Candace Savonen](https://www.cansavvy.com/) |
| Package Developers ([Leanbuild](https://github.com/jhudsl/leanbuild)) | [John Muschelli](https://johnmuschelli.com/), [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/) |
| **Art and Design** |  |
| Illustrator(s) | Created graphics for the course |
| Figure Artist(s) | Created figures/plots for course |
| Videographer(s) | Filmed videos |
| Videography Editor(s) | Edited film |
| Audiographer(s) | Recorded audio |
| Audiography Editor(s) | Edited audio recordings |
| **Funding** |  |
| Funder(s) | Institution/individual who funded course including grant number |
| Funding Staff | Staff members who help with funding |

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value   
## version R version 4.0.2 (2020-06-22)  
## os Ubuntu 20.04.3 LTS   
## system x86\_64, linux-gnu   
## ui X11   
## language (EN)   
## collate en\_US.UTF-8   
## ctype en\_US.UTF-8   
## tz Etc/UTC   
## date 2021-11-17   
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date lib source   
## assertthat 0.2.1 2019-03-21 [1] RSPM (R 4.0.3)   
## backports 1.1.10 2020-09-15 [1] RSPM (R 4.0.2)   
## bookdown 0.24 2021-11-03 [1] Github (rstudio/bookdown@88bc4ea)   
## callr 3.4.4 2020-09-07 [1] RSPM (R 4.0.2)   
## cli 2.0.2 2020-02-28 [1] RSPM (R 4.0.0)   
## crayon 1.3.4 2017-09-16 [1] RSPM (R 4.0.0)   
## desc 1.2.0 2018-05-01 [1] RSPM (R 4.0.3)   
## devtools 2.3.2 2020-09-18 [1] RSPM (R 4.0.3)   
## digest 0.6.25 2020-02-23 [1] RSPM (R 4.0.0)   
## ellipsis 0.3.1 2020-05-15 [1] RSPM (R 4.0.3)   
## evaluate 0.14 2019-05-28 [1] RSPM (R 4.0.3)   
## fansi 0.4.1 2020-01-08 [1] RSPM (R 4.0.0)   
## fs 1.5.0 2020-07-31 [1] RSPM (R 4.0.3)   
## glue 1.4.2 2020-08-27 [1] RSPM (R 4.0.3)   
## htmltools 0.5.0 2020-06-16 [1] RSPM (R 4.0.1)   
## knitr 1.33 2021-11-03 [1] Github (yihui/knitr@a1052d1)   
## lifecycle 1.0.0 2021-02-15 [1] CRAN (R 4.0.2)   
## magrittr 1.5 2014-11-22 [1] RSPM (R 4.0.0)   
## memoise 1.1.0 2017-04-21 [1] RSPM (R 4.0.0)   
## pkgbuild 1.1.0 2020-07-13 [1] RSPM (R 4.0.2)   
## pkgload 1.1.0 2020-05-29 [1] RSPM (R 4.0.3)   
## prettyunits 1.1.1 2020-01-24 [1] RSPM (R 4.0.3)   
## processx 3.4.4 2020-09-03 [1] RSPM (R 4.0.2)   
## ps 1.3.4 2020-08-11 [1] RSPM (R 4.0.2)   
## purrr 0.3.4 2020-04-17 [1] RSPM (R 4.0.3)   
## R6 2.4.1 2019-11-12 [1] RSPM (R 4.0.0)   
## remotes 2.2.0 2020-07-21 [1] RSPM (R 4.0.3)   
## rlang 0.4.10 2021-11-03 [1] Github (r-lib/rlang@f0c9be5)   
## rmarkdown 2.10 2021-11-03 [1] Github (rstudio/rmarkdown@02d3c25)  
## rprojroot 1.3-2 2018-01-03 [1] RSPM (R 4.0.0)   
## sessioninfo 1.1.1 2018-11-05 [1] RSPM (R 4.0.3)   
## stringi 1.5.3 2020-09-09 [1] RSPM (R 4.0.3)   
## stringr 1.4.0 2019-02-10 [1] RSPM (R 4.0.3)   
## testthat 3.0.1 2021-11-03 [1] Github (R-lib/testthat@e99155a)   
## usethis 2.1.3.9000 2021-11-03 [1] Github (r-lib/usethis@dde4610)   
## withr 2.3.0 2020-09-22 [1] RSPM (R 4.0.2)   
## xfun 0.26 2021-11-03 [1] Github (yihui/xfun@74c2a66)   
## yaml 2.2.1 2020-02-01 [1] RSPM (R 4.0.3)   
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library