minor hyperlink change

August authored 17 minutes ago

055071e0

README.md 26.4 KB

PanGIA_GUI_README

Gus Thomas 9/28/2021

PanGIA GUI Manual:

PanGIA GUI Insignia Image HERE

Your Guide to the PanGIA GUI

Table of Contents

Section 1): Introduction

This report contains instructions for the installation and use of the graphical interface designed for PanGIA. PanGIA is standalone and may be run independently from the command line. However, this forces the responsibilities of manual configuration and record keeping onto the user.

This GUI companion to PanGIA makes the pathway to discovery more efficient and exciting by streamlining the run specification/execution process, providing a robust job queue, enabling real-time PanGIA processing, and constructing an intuitive, informative, dynamic visualization tool for result analysis. The GUI is composed of six parent pages: Dashboard, Users, Settings, Projects, PanGIA, and Real Time. If you have already completed PanGIA setup and are using this document as a GUI reference, please consult **Section 4.2** onward.

Section 2): Initial Download

This section details the process of downloading PanGIA, the GUI, and curated reference databases necessary to run PanGIA.

Section 2.1): PanGIA and PanGIA GUI

Section 2.1.1): Downloading from the Command Line

1. Navigate to preferred local path.

2. Use a `git clone` command on `git@gitlab.mriglobal.org:gus-mri/pangia_flask_gui_docker.git` - navigate into the cloned repository to verify.

Section 2.1.2): Downloading Manually

1. Go to http://gitlab.mriglobal.org/gus-mri/pangia_flask_gui_docker and download the repo. Extract files to your preferred location. Navigate into the extracted repository to verify.

Section 2.2): Reference Databases

Talk about ftp download here. Check with the team about whether the new database has been indexed/tested and is accessible.

Section 3): Preparing the Compute Environment

PanGIA requires a specific compute environment to satisfy its run-time dependencies. We consider two cases: an environment constructed locally within a specific Linux distribution, and a PanGIA-friendly Docker environment independent of the host machine.

Section 3.1): Preparing a Non-Dockerized Environment

1. Install the latest distribution of Anaconda or miniconda. Ensure you are running a compatible Linux distribution (Ubuntu/Pop-OS 16.04-20.04).

2. In a terminal, navigate to the PanGIA GUI directory (named `pangia_flask_gui_docker` by default) created in **Section 2.1** and use command: `conda env create –f gui.yml`. This command acts on the .yml file within the GUI root directory to construct a `conda` environment. By default, the name of the environment will be `gui`, but you may replace this with any preferred name by changing the prefix of the .yml file prior to creating the environment.

3. Open a command line window and navigate to the GUI directory. Use command: `conda activate XXX`, where `XXX` is the name of the environment. Use the following series of commands to finish initial database and environment setup:

- `export FLASK_APP=pangia_gui.py`
- `flask db init`
- `flask db migrate –m`
- `flask db upgrade`

Ignore any errors that pop up! These commands are used to initialize the flask database that the GUI relies on. In the event that you reset your GUI (see below), having entered these commands will mitigate any further action you need to take.

Whenever accessing the GUI, **begin by opening three separate command line terminals**, and navigate all three into the `/pangia_flask_gui_docker/gui` directory. In each of the three windows, execute the `conda activate XXX` command, where `XXX` is the name of the `conda` environment specified above in **Section 3.1** (`gui` by default). Note that each instance will be responsible for running one of three services comprising the GUI. See **Section 4.1** for more information. **The GUI can be completely reset by navigating to http://localhost:5000/auth/reset_site - this is recommended only in cases where the job queue is stuck. Bugfix pending.**

Section 3.2): Preparing a Dockerized Environment

1. Download Docker and Docker Compose. Docker Compose must be installed separately, if you are using any Linux OS. Please follow Docker's official instructions at https://docs.docker.com/get-docker/ to install Docker - for Linux machines, an additional step is required to grant root

access to non-root users for Docker commands. Without taking this step, you will need to prepend every Docker command with `sudo`. Please see https://docs.docker.com/engine/install/linux-postinstall/ for instructions. If you are using a Linux OS and would like access to a Docker GUI (in lieu of the Linux-incompatible Docker Desktop), we recommend the lazydocker package. See https://github.com/jesseduffield/lazydocker for details.

2. On the command line, navigate to `/pangia_flask_gui_docker`.

3. In the sub-directory `/pangia_flask_gui_docker/PanGIA`, add folders `uploads` and `database` if they aren't there already. Transfer any databases you previously downloaded in **Section 2.2** into the `database` folder. The GUI will place your Projects (which will contain relevant user-uploaded fastq files) into the `uploads` folder. See **Section 4.2.4** for more details about Projects.

4. On the command line at `/pangia_flask_gui_docker` use command: `DOCKER_BUILDKIT=1 docker-compose up --build`. Please wait while the required Docker images are pulled from Docker Hub, and the `conda` environments are constructed. This may take a few minutes, but subsequently launching the service will take seconds. **Section 4.1.2** describes how to launch the service in both regular-mode and detached-mode after initial setup.

5. **Navigate to http://localhost:5000/auth/reset_site for first-time GUI access**. After first-time access, navigate to `localhost:5000`. If you bring down the containers (via computer shutdown or a `docker-compose down` command), use `docker-compose up` to bring the built service back up.

Section 4): Using the GUI

Running PanGIA through the GUI assumes that three independent services – the worker, scheduler, and Flask server – are already operational. **Section 4.1** covers the process of bringing each service online if you are not using Docker. We will also describe the Dockerized case, in which much of the legwork is handled for you **Section 3.2** by Docker Compose.

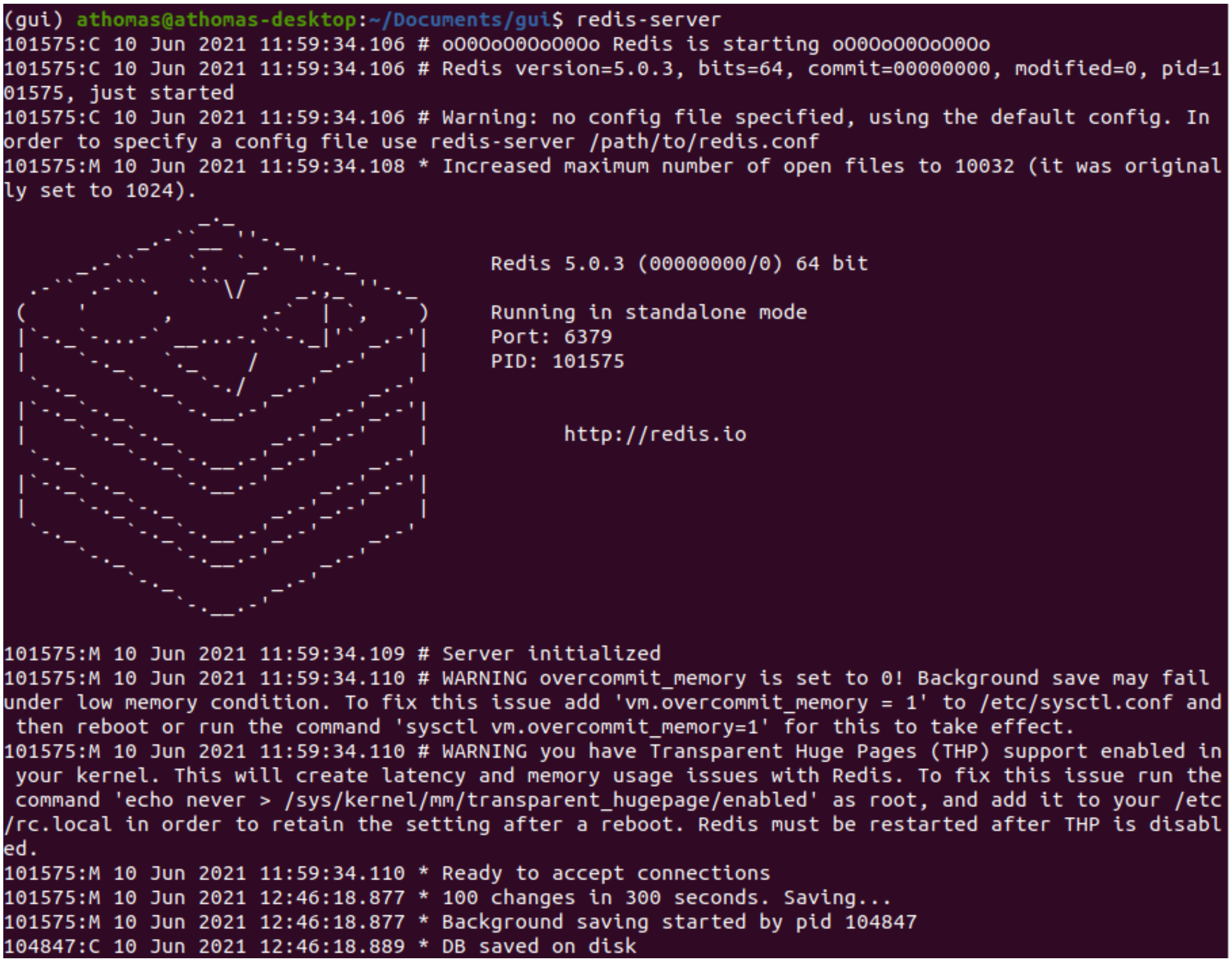
A complete tour of the GUI can be found in **Section 4.2**. We describe two examples of PanGIA runs in **Section 4.3**; a run using standard analysis in **Section 4.3.1**, and a Real-Time run in **Section 4.3.2**.

Section 4.1): Accessing the GUI

Section 4.1.1): *Non-Dockerized Case*

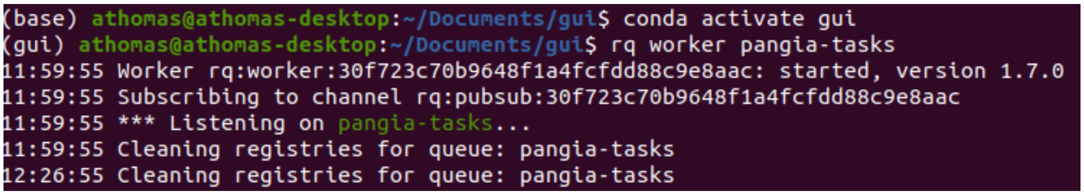
1. Following instructions listed at the the end of **Section 3.1**, prepare three command-line terminals; each must be navigated to `/pangia_flask_gui_docker` and an activated `conda` environment.

2. In the first window use command: `redis-server`. The window should look something like this:



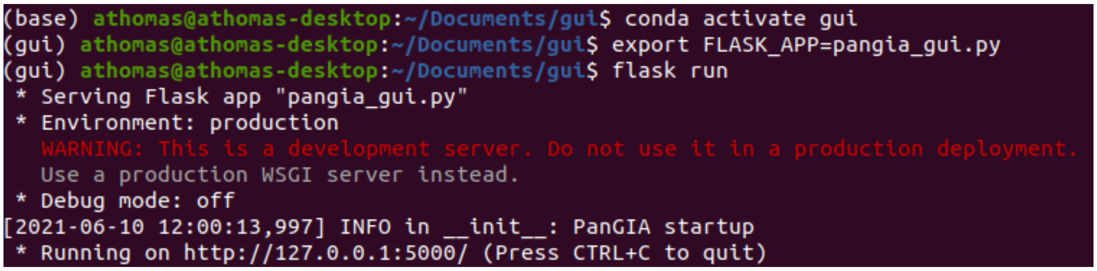
```
(gui) athomas@athomas-desktop:~/Documents/gui$ redis-server
101575:C 10 Jun 2021 11:59:34.106 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
101575:C 10 Jun 2021 11:59:34.106 # Redis version=5.0.3, bits=64, commit=00000000, modified=0, pid=1
01575, just started
101575:C 10 Jun 2021 11:59:34.106 # Warning: no config file specified, using the default config. In
order to specify a config file use redis-server /path/to/redis.conf
101575:M 10 Jun 2021 11:59:34.108 * Increased maximum number of open files to 10032 (it was original
ly set to 1024).
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 5.0.3 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in standalone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 101575
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

101575:M 10 Jun 2021 11:59:34.109 # Server initialized
101575:M 10 Jun 2021 11:59:34.110 # WARNING overcommit_memory is set to 0! Background save may fail
under low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and
 then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
101575:M 10 Jun 2021 11:59:34.110 # WARNING you have Transparent Huge Pages (THP) support enabled in
 your kernel. This will create latency and memory usage issues with Redis. To fix this issue run the
 command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc
/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabl
ed.
101575:M 10 Jun 2021 11:59:34.110 * Ready to accept connections
101575:M 10 Jun 2021 12:46:18.877 * 100 changes in 300 seconds. Saving...
101575:M 10 Jun 2021 12:46:18.877 * Background saving started by pid 104847
104847:C 10 Jun 2021 12:46:18.889 * DB saved on disk
```

3. In the second window use command: `rq worker pangia-tasks` . The window should look something like this:



```
(base) athomas@athomas-desktop:~/Documents/gui$ conda activate gui
(gui) athomas@athomas-desktop:~/Documents/gui$ rq worker pangia-tasks
11:59:55 Worker rq:worker:30f723c70b9648f1a4fcfdd88c9e8aac: started, version 1.7.0
11:59:55 Subscribing to channel rq:pubsub:30f723c70b9648f1a4fcfdd88c9e8aac
11:59:55 *** Listening on pangia-tasks...
11:59:55 Cleaning registries for queue: pangia-tasks
12:26:55 Cleaning registries for queue: pangia-tasks
```

4. In the third window use command: `flask run` . The window should look something like this:



```
(base) athomas@athomas-desktop:~/Documents/gui$ conda activate gui
(gui) athomas@athomas-desktop:~/Documents/gui$ export FLASK_APP=pangia_gui.py
(gui) athomas@athomas-desktop:~/Documents/gui$ flask run
 * Serving Flask app "pangia_gui.py"
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
[2021-06-10 12:00:13,997] INFO in __init__: PanGIA startup
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

5. Click on the link displayed at the bottom of the `flask run` terminal. Alternatively, in a web browser, navigate to localhost:5000 for GUI access.

Section 4.1.2): Dockerized Case

1. In any command line window, navigate to the `/pangia_flask_gui_docker` directory.

2. Use command: `docker-compose up` . **If you would like to be able to close the terminal window and allow the Docker service to continue running, instead use command: `docker-compose up --detach` . Our container will restart automatically, unless it is explicitly stopped**

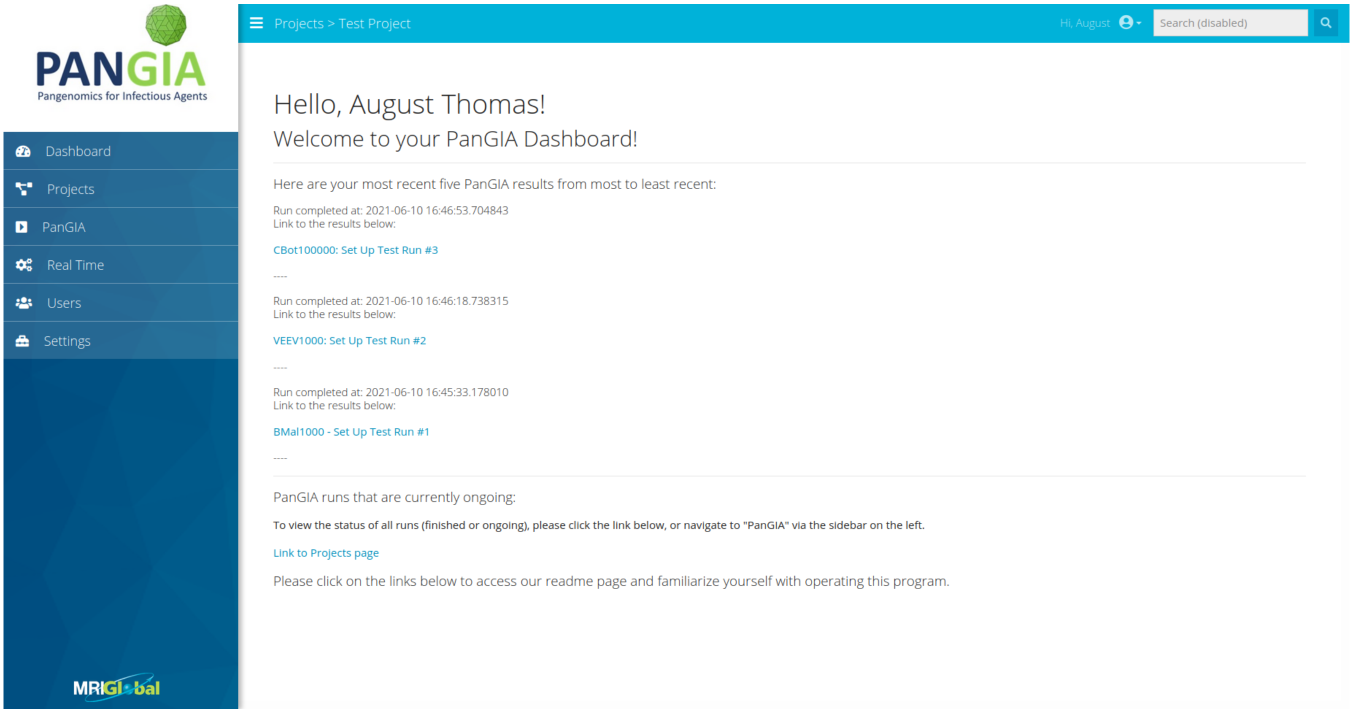
with the command `docker-compose down`.

3. Click on the link displayed at the bottom of the terminal. Alternatively, in a web browser, navigate to localhost:5000 for GUI access.

Section 4.2): GUI Navigation

Section 4.2.1): Dashboard

After logging in, users are directed to this page. It lists the history of both the most recent PanGIA runs and ongoing runs. Links to results and/or the ongoing job status page is provided under each listing. At the very bottom of the page, users may click a link directing to the PanGIA section of the GUI, described in more detail below.



Hello, August Thomas!

Welcome to your PanGIA Dashboard!

Here are your most recent five PanGIA results from most to least recent:

Run completed at: 2021-06-10 16:46:53.704843
Link to the results below:

CBot100000: Set Up Test Run #3

Run completed at: 2021-06-10 16:46:18.738315
Link to the results below:

VEEV1000: Set Up Test Run #2

Run completed at: 2021-06-10 16:45:33.178010
Link to the results below:

BMal1000 - Set Up Test Run #1

PanGIA runs that are currently ongoing:

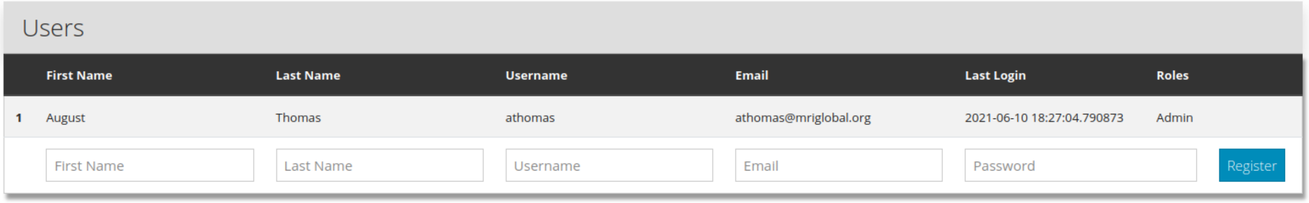
To view the status of all runs (finished or ongoing), please click the link below, or navigate to "PanGIA" via the sidebar on the left.

Link to Projects page

Please click on the links below to access our readme page and familiarize yourself with operating this program.

Section 4.2.2): Users

Interface for adding users to the GUI's persisted database. Current users are listed with name, username, email, date of last login, and role (Admin, User, etc.).



Users

	First Name	Last Name	Username	Email	Last Login	Roles
1	August	Thomas	athomas	athomas@mriglobal.org	2021-06-10 18:27:04.790873	Admin
	First Name	Last Name	Username	Email	Password	Register

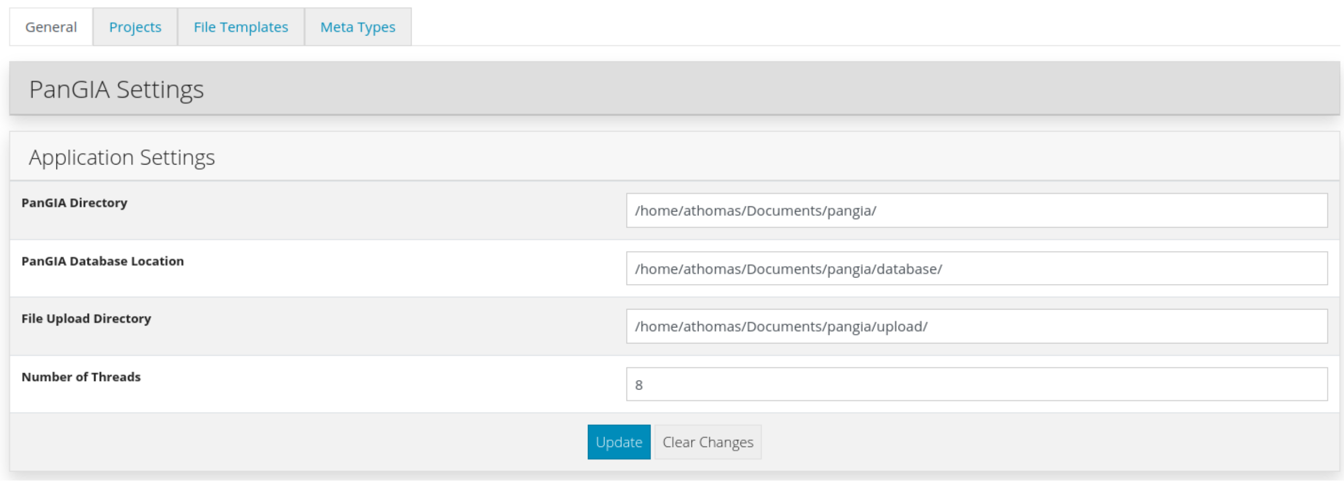
Section 4.2.3): Settings

Multi-tabbed interface for specification of general settings, PanGIA preprocessing/run parameters, project categories, file templates, and meta-types. Specifics for each tab and sub-tab are listed below.

Section 4.2.3.1): General Settings

These settings are dedicated to tailoring PanGIA runs to the user's default needs. Most general settings may be adjusted on-the-fly, while preparing any PanGIA run. However, preparing defaults here will save time. The general settings are subdivided into three categories:

1. **Application Settings**: User specifies PanGIA directory path and PanGIA database/uploads path. If you are running the GUI without Docker, you may need to adjust the path to reflect wherever you put the `/pangia_flask_gui_docker/PanGIA` folder. **If you are using Docker, this pathway will already reference the PanGIA directory mounted to the service - you should not need to make any changes**. Users may also adjust thread-count made available for PanGIA. An even number is recommended. Default is two threads.

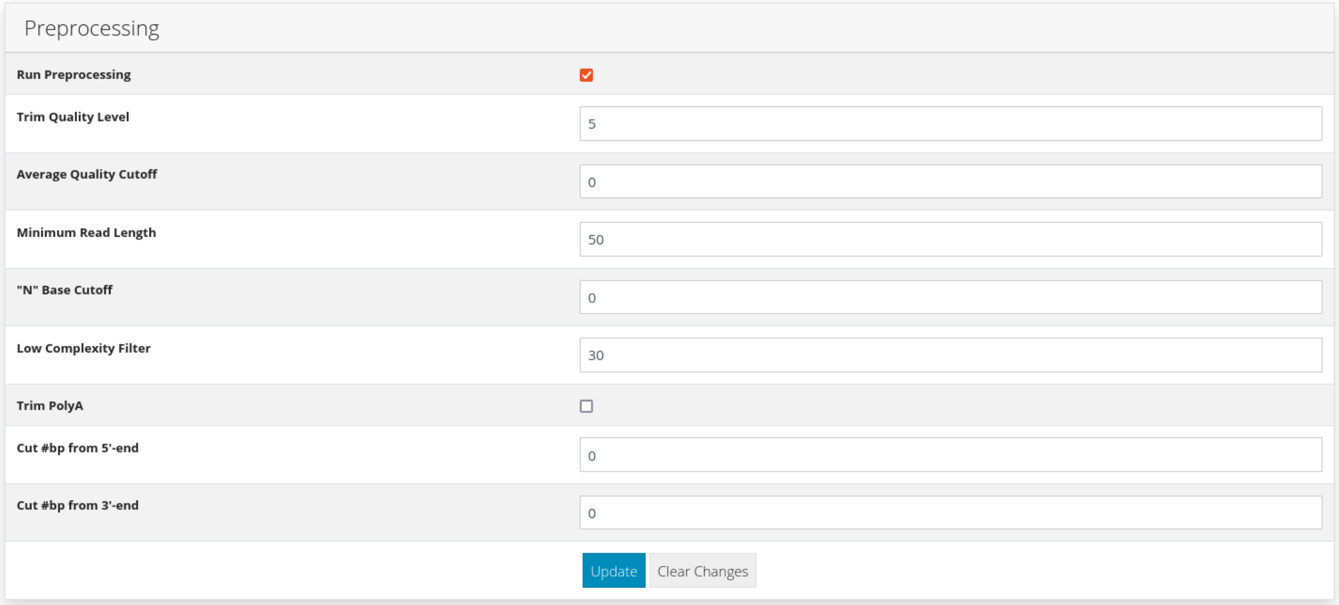


PanGIA Settings

Application Settings

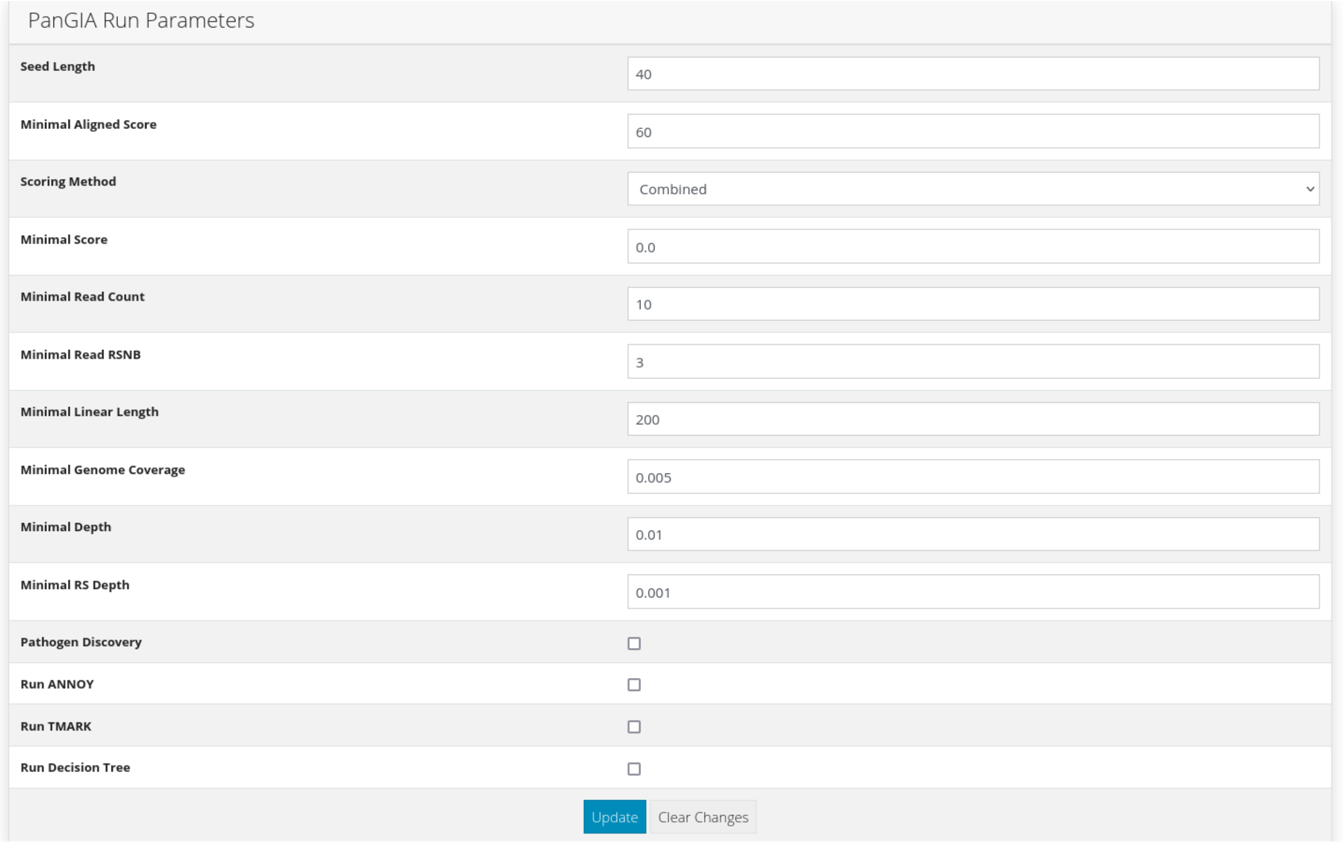
PanGIA Directory	/home/athomas/Documents/pangia/
PanGIA Database Location	/home/athomas/Documents/pangia/database/
File Upload Directory	/home/athomas/Documents/pangia/upload/
Number of Threads	8

2. **Preprocessing**: Options to alter/filter reads prior to PanGIA analysis. May trim reads for quality level, adjust the average quality, minimum read length, and "N" base cutoff values, apply a low complexity filter, exclude PolyA tails, or indicate a custom number of base pairs to cut from 5' or 3' ends of reads. The user may opt to skip preprocessing (equivalent to running PanGIA on the command line with the `-da` tag) by setting all values to zero and unchecking all boxes.


Preprocessing

Run Preprocessing	☑
Trim Quality Level	5
Average Quality Cutoff	0
Minimum Read Length	50
"N" Base Cutoff	0
Low Complexity Filter	30
Trim PolyA	□
Cut #bp from 5'-end	0
Cut #bp from 3'-end	0

3. **Run Parameters**: Options determining read scoring, and additional forms of analysis (TMARK, Decision Tree, etc.). You may specify values for seed length, and assign minimal values for score, aligned score, read count, read RSNB, linear length, percent genome coverage, and depth/RS depth. Please note that the current version of the GUI Visualizer will not take values entered here into account on launch; all Slider widgets are set to launch with a default value of 0. The scoring method for the PanGIA run may be selected from a drop-down menu (Standalone/Background/Combined), and a series of check boxes determine if additional analyses will be included in the run.

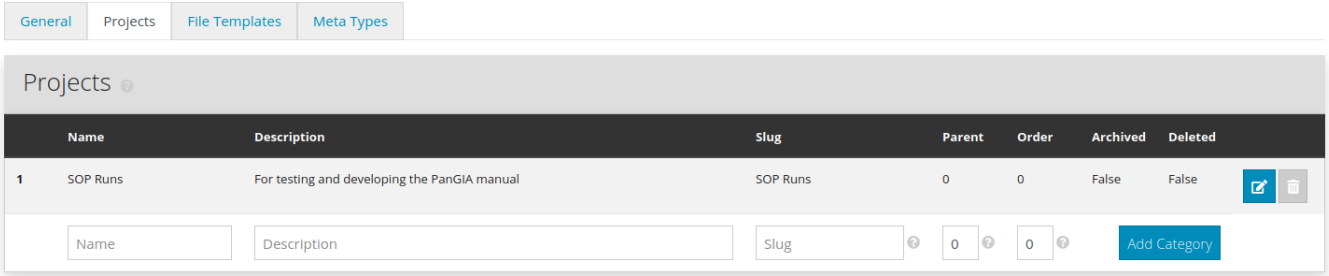


PanGIA Run Parameters

Seed Length	40
Minimal Aligned Score	60
Scoring Method	Combined
Minimal Score	0.0
Minimal Read Count	10
Minimal Read RSNB	3
Minimal Linear Length	200
Minimal Genome Coverage	0.005
Minimal Depth	0.01
Minimal RS Depth	0.001
Pathogen Discovery	□
Run ANNOY	□
Run TMARK	□
Run Decision Tree	□

Section 4.2.3.2): Project Settings

Projects are used to consolidate and organize related .fastq files and PanGIA runs. Existing *Projects* may be archived or deleted. *Projects* are instantiated with a name, description, and slug. They may be constructed as children of an existing parent *Project*. When providing a name and slug, use only alphanumeric and '_' characters. **Please use the same value for both 'name' and 'slug' - bugfix pending.**

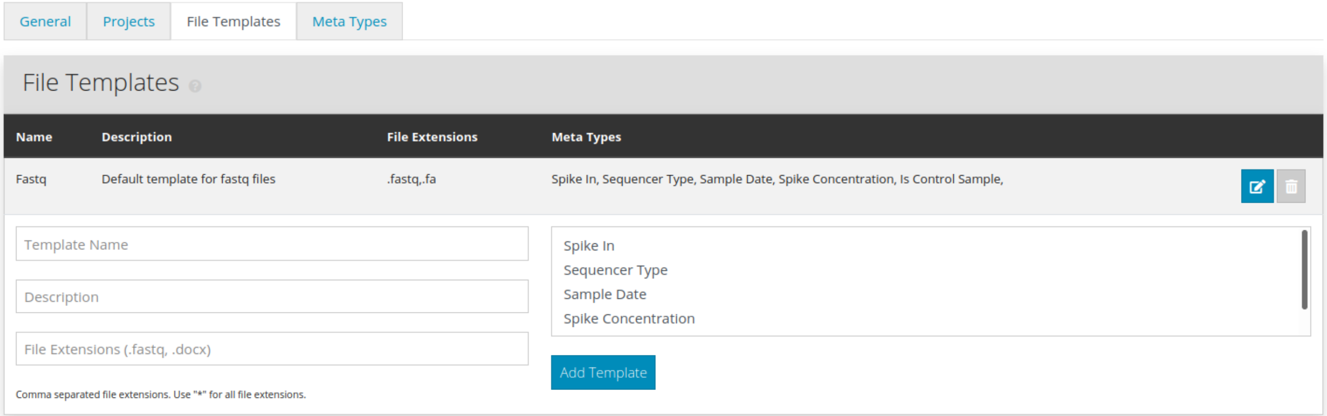


Projects

	Name	Description	Slug	Parent	Order	Archived	Deleted
1	SOP Runs	For testing and developing the PanGIA manual	SOP Runs	0	0	False	False
	Name	Description	Slug	0	0		

Section 4.2.3.3): File Template Settings

User may specify PanGIA input file types other than .fastq on this tab. File Templates require a name, description, and file extension. When creating a new File Template, the user must decide which Meta-Types will be available for runs using that File Template. A default for .fastq files is included.

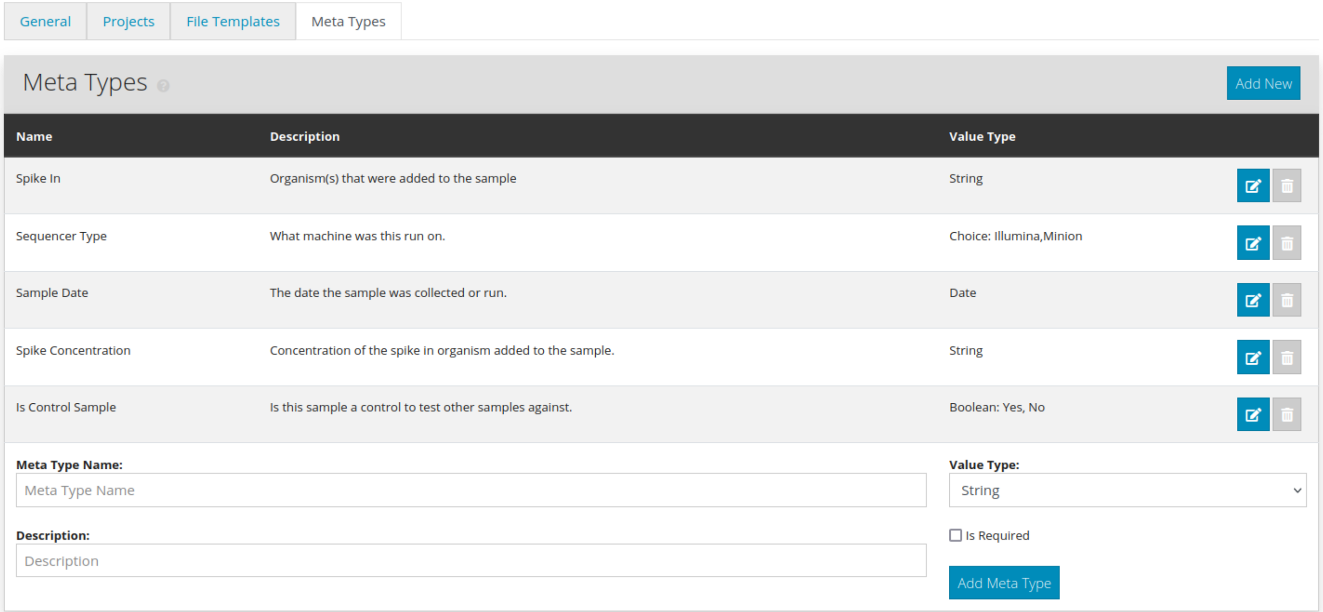


Name	Description	File Extensions	Meta Types
Fastq	Default template for fastq files	.fastq,.fa	Spike In, Sequencer Type, Sample Date, Spike Concentration, Is Control Sample,

Comma separated file extensions. Use "*" for all file extensions.

Section 4.2.3.4): Meta-Type Settings

User may impose additional rules regarding meta-data, grouping it together and limiting input variance. Default Meta-Types include names of spiked-in organisms and spike-concentration, sequencing protocol, and the sampling date. User may specify whether data was a control sample. New Meta-Types require a name, description, and value (string, integer, Boolean, date, etc.), and may be designated as 'required' if desired.



Name	Description	Value Type
Spike In	Organism(s) that were added to the sample	String
Sequencer Type	What machine was this run on.	Choice: Illumina,Minion
Sample Date	The date the sample was collected or run.	Date
Spike Concentration	Concentration of the spike in organism added to the sample.	String
Is Control Sample	Is this sample a control to test other samples against.	Boolean: Yes, No

Meta Type Name: Meta Type Name

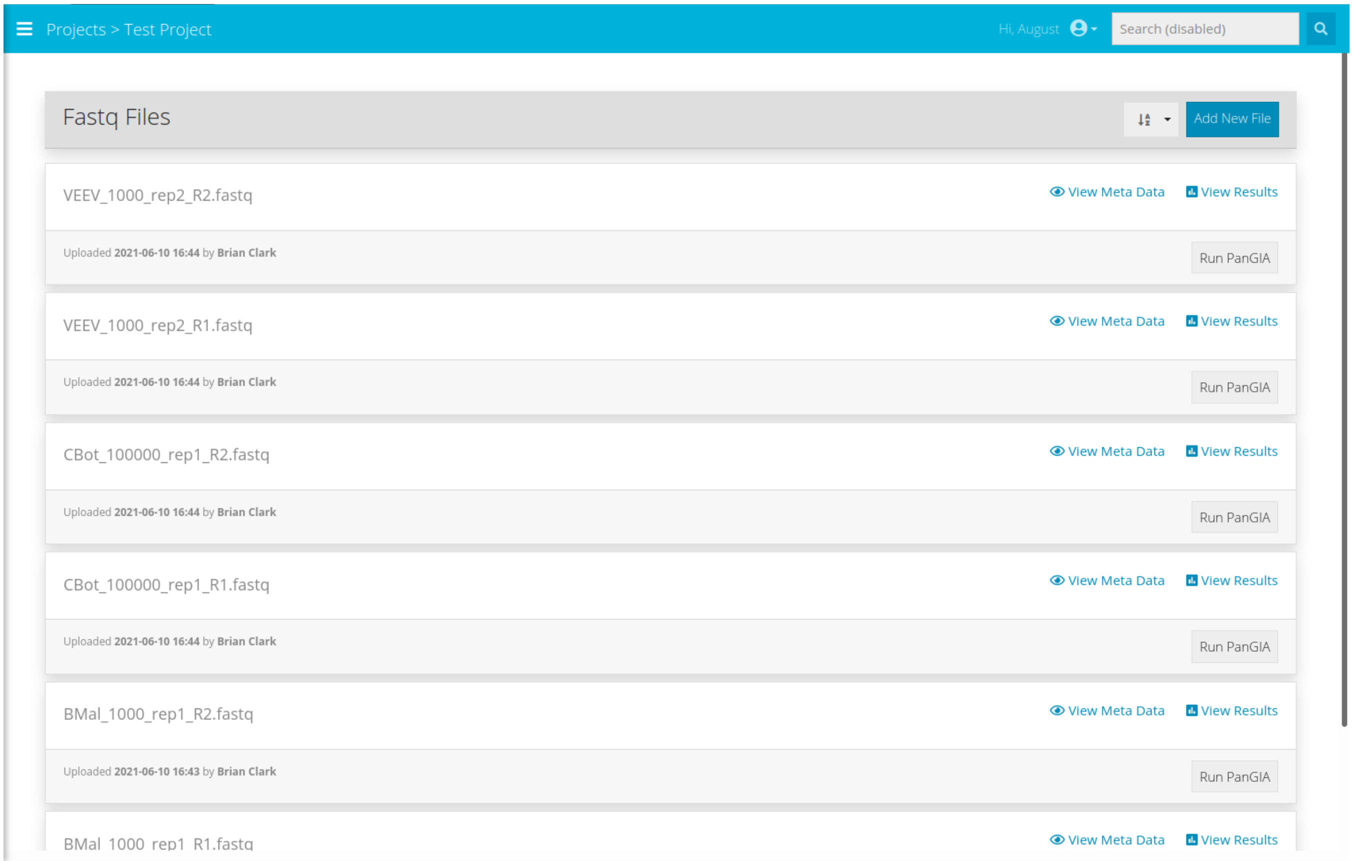
Value Type: String

Description: Description

☐ Is Required

Section 4.2.4): Projects

Provides the name of each *Project* constructed in the Settings. Clicking on the name of a *Project* yields a list of all the files that have been uploaded to it. The list is grouped by *File Template*. Files may be ordered by ascending/descending or date/name. Each *File Template* header has a large blue button labeled 'Add New File' - this directs the user to the upload interface. Files uploaded through the GUI are copied into a new directory named after the *Project* in the `/uploads` directory.



Fastq Files

VEEV_1000_rep2_R2.fastq
Uploaded **2021-06-10 16:44** by **Brian Clark**

VEEV_1000_rep2_R1.fastq
Uploaded **2021-06-10 16:44** by **Brian Clark**

CBot_100000_rep1_R2.fastq
Uploaded **2021-06-10 16:44** by **Brian Clark**

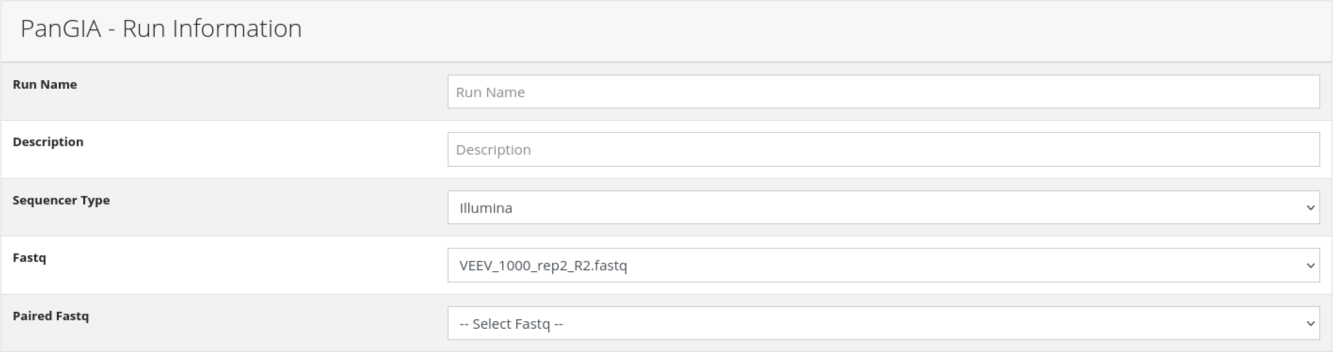
CBot_100000_rep1_R1.fastq
Uploaded **2021-06-10 16:44** by **Brian Clark**

BMal_1000_rep1_R2.fastq
Uploaded **2021-06-10 16:43** by **Brian Clark**

BMal_1000_rep1_R1.fastq

Section 4.2.4.1): Running PanGIA from the Projects Tab

To start a job from the *Project* menu, use the 'Run PanGIA' button on any file uploaded to the *Project*. Each run requires a name and description. Drop-down menus specify which paired files are to be used. The first menu option defaults to whatever file the Run Information page was accessed from. All General settings may be adjusted here: the interface is identical to that appearing in **Section 4.2.3.1**.



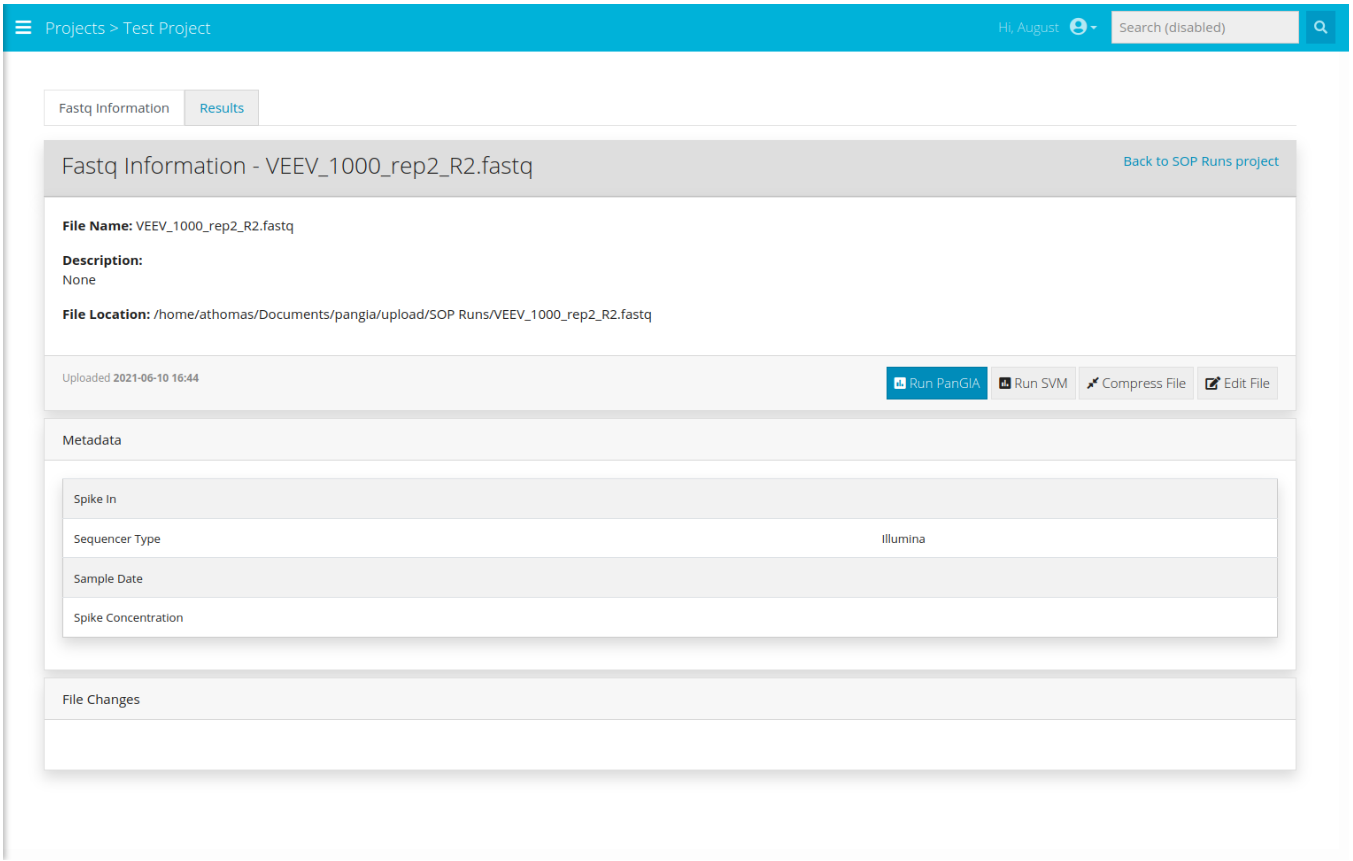
PanGIA - Run Information

Run Name	Run Name
Description	Description
Sequencer Type	Illumina
Fastq	VEEV_1000_rep2_R2.fastq
Paired Fastq	-- Select Fastq --

***Section 4.2.4.2)*: Exploring the Projects Tab**

Files listed within a *Project* have three interactive links. The 'View Meta Data' and 'View Results' links direct the user to a new page, with two tabs. The first tab is named 'X Information', where 'X' is the *File Template*; the second is named *Results*.

1. The 'X Information' tab ('Fastq Information' in the image below) specifies the filename, upload date, filepath, and description.



Fastq Information - VEEV_1000_rep2_R2.fastq

File Name: VEEV_1000_rep2_R2.fastq

Description:
None

File Location: /home/athomas/Documents/pangia/upload/SOP Runs/VEEV_1000_rep2_R2.fastq

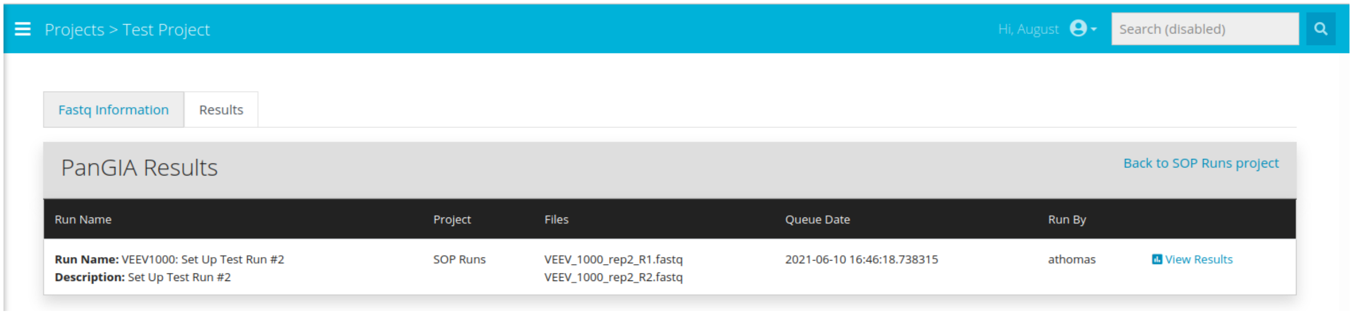
Uploaded **2021-06-10 16:44**

Metadata

Spike In	
Sequencer Type	Illumina
Sample Date	
Spike Concentration	

File Changes

2. The ‘Results’ tab lists all PanGIA runs that the file has been part of. Details include run name/date, description, parent *Project*, user, and filenames.



PanGIA Results

Run Name	Project	Files	Queue Date	Run By	
Run Name: VEEV1000: Set Up Test Run #2 **Description:** Set Up Test Run #2	SOP Runs	VEEV_1000_rep2_R1.fastq VEEV_1000_rep2_R2.fastq	2021-06-10 16:46:18.738315	athomas	View Results

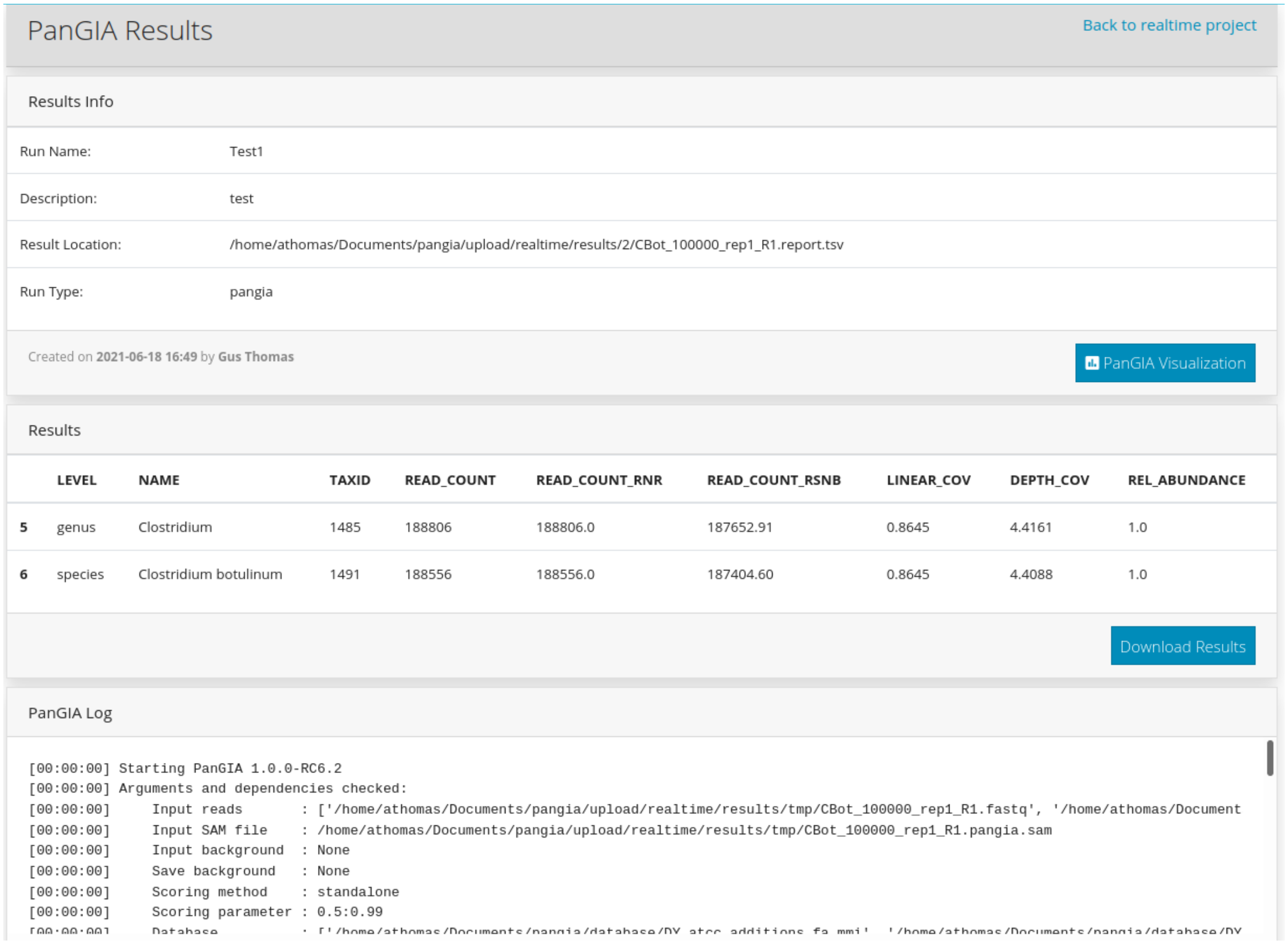
3. Notice above: on the right side, runs have a link labeled ‘View Results’. Following this link directs the user to the PanGIA *Results* section of the GUI, described below in **Section 4.2.5**.

Section 4.2.5): PanGIA & PanGIA Results

Lists all queued and completed jobs, regardless of Project. Each completed job contains the same information found in the ‘Results’ tab accessible through the *Projects* section of the GUI; each job also has the same ‘View Results’ link. Both are described above in **Section 4.2.4**.

Section 4.2.5.1): Results Page

The *Results* page provides run name and description, and features an interactive window to browse rows of the .tsv file output of the PanGIA run. This page also links to the GUI Visualizer, which will graphically display that output. An in-depth PanGIA run log is provided at the bottom – the printout there is similar to what can be found in in the worker printout during the run. An example *Results* page appears below.



Results Info

Run Name:	Test1
Description:	test
Result Location:	/home/athomas/Documents/pangia/upload/realtime/results/2/CBot_100000_rep1_R1.report.tsv
Run Type:	pangia

Created on **2021-06-18 16:49** by **Gus Thomas**

Results

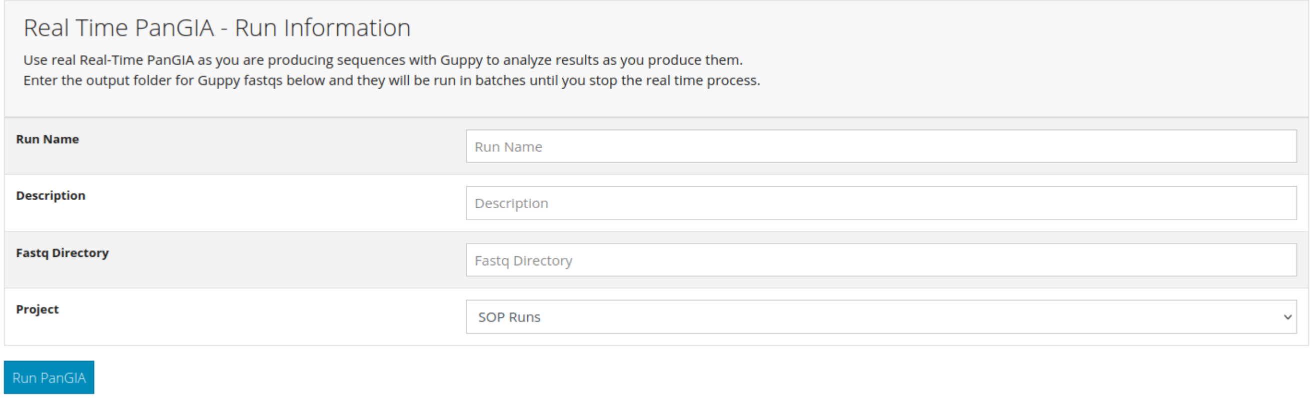
	LEVEL	NAME	TAXID	READ_COUNT	READ_COUNT_RNR	READ_COUNT_RSNB	LINEAR_COV	DEPTH_COV	REL_ABUNDANCE
5	genus	Clostridium	1485	188806	188806.0	187652.91	0.8645	4.4161	1.0
6	species	Clostridium botulinum	1491	188556	188556.0	187404.60	0.8645	4.4088	1.0

PanGIA Log

```
[00:00:00] Starting PanGIA 1.0.0-RC6.2
[00:00:00] Arguments and dependencies checked:
[00:00:00]     Input reads       : ['/home/athomas/Documents/pangia/upload/realtime/results/tmp/CBot_100000_rep1_R1.fastq', '/home/athomas/Document
[00:00:00]     Input SAM file    : /home/athomas/Documents/pangia/upload/realtime/results/tmp/CBot_100000_rep1_R1.pangia.sam
[00:00:00]     Input background  : None
[00:00:00]     Save background   : None
[00:00:00]     Scoring method    : standalone
[00:00:00]     Scoring parameter : 0.5:0.99
[00:00:00]     Database          : ['/home/athomas/Documents/pangia/database/DY_atcc_additions.fa.mmi', '/home/athomas/Documents/pangia/database/DY
```

Section 4.2.6): Real-Time PanGIA

User specifies a job name, description, and associated *Project*, as usual. Instead of paired files, the user must provide the path to a directory that Real-Time PanGIA is listening for. This directory should be the dumping point for ongoing nanopore sequencing, processed by Guppy. Notably, once Real-Time PanGIA is initiated, the user must manually terminate the job as desired. Completed Real-Time jobs persist PanGIA results from the last job iteration.



Real Time PanGIA - Run Information

Use real Real-Time PanGIA as you are producing sequences with Guppy to analyze results as you produce them.
Enter the output folder for Guppy fastqs below and they will be run in batches until you stop the real time process.

Run Name	Run Name
Description	Description
Fastq Directory	Fastq Directory
Project	SOP Runs

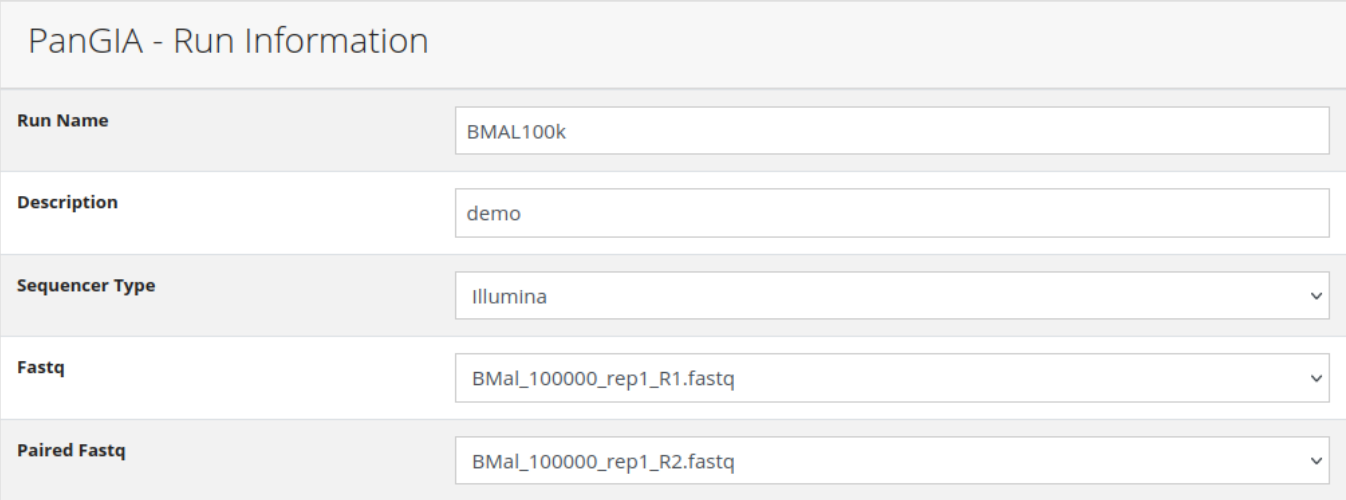
IMPORTANT: **If you are using the Dockerized version of the GUI, the fastq directory path must always begin with `/gui_flask`**. This is because `/gui_flask` is the name of the Docker “volume” - it is the version of the repository on the local file system (`/pangia_flask_gui_docker` by default) that the Docker service has access to. We have included a folder for Real-Time output that you can use: `/PanGIA/real_time` . We suggest that you set Guppy output to dump into this folder, and also specify it here. In that case, the full fastq directory path would be: **`/gui_flask/PanGIA/real_time`** .

Section 4.3): Examples of Usage

Section 4.3.1): Baseline Example

The following walkthrough describes a typical PanGIA run with default settings. For this example, assume the Project 'SOP_Runs' has already been established. For the purposes of this tutorial, assume that two .fastq files – named 'BMal_100000_rep1_R1' and 'BMal_100000_rep1_R2' – have already been uploaded to the Project. The file page for such a scenario appears below.

Clicking on the Run PanGIA buttons associated with either .fastq file directs to the Run PanGIA page. As discussed in **Section 4.2.4.1**, this page is split into Run Information and General settings. Pressing the button associated with 'BMal_100000_rep1_R1' directs to the Run Information page shown below.

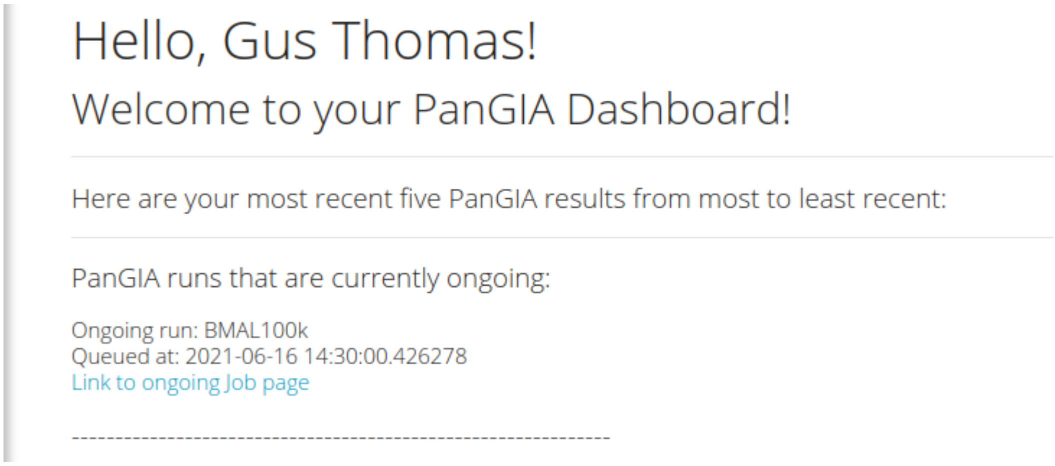


PanGIA - Run Information

Run Name	BMAL100k
Description	demo
Sequencer Type	Illumina
Fastq	BMal_100000_rep1_R1.fastq
Paired Fastq	BMal_100000_rep1_R2.fastq

This run has been given a name and description, and a selection for the 'Paired Fastq' has been made – in this case, 'BMal_100000_rep1_R2'. None of the General settings on the Run PanGIA page need to be adjusted, as this is a default PanGIA run – so this job is ready to be executed by clicking on the Run PanGIA button at the bottom of the page!

The PanGIA job is now running, which may be verified at the Dashboard or the Running Job page. If additional PanGIA runs were queued behind 'BMAL100k', they would be listed in the Dashboard.



Hello, Gus Thomas!

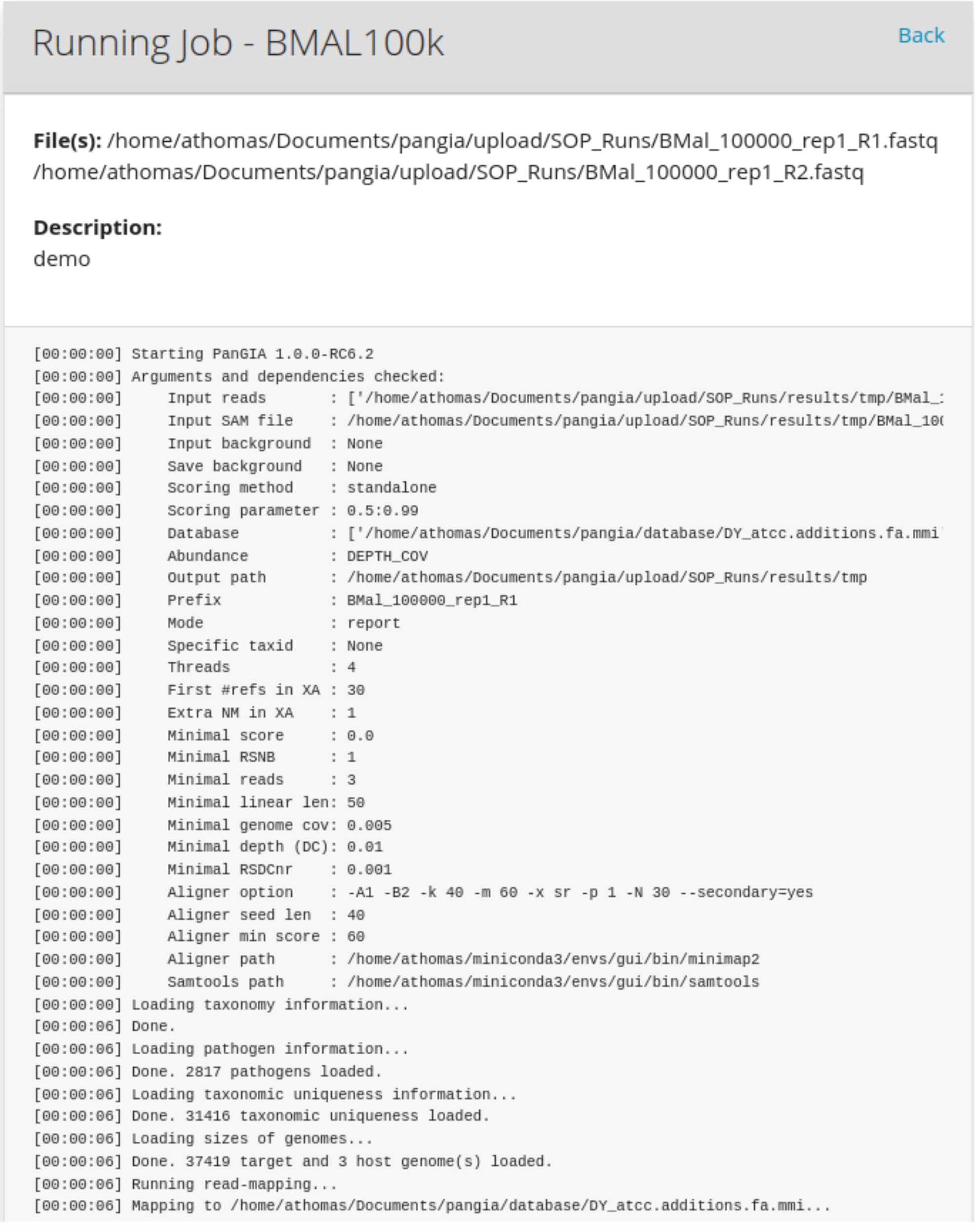
Welcome to your PanGIA Dashboard!

Here are your most recent five PanGIA results from most to least recent:

PanGIA runs that are currently ongoing:

Ongoing run: BMAL100k
Queued at: 2021-06-16 14:30:00.426278
Link to ongoing Job page

The Running Job page is a run log, and should look similar to the rq worker printout, as seen below:



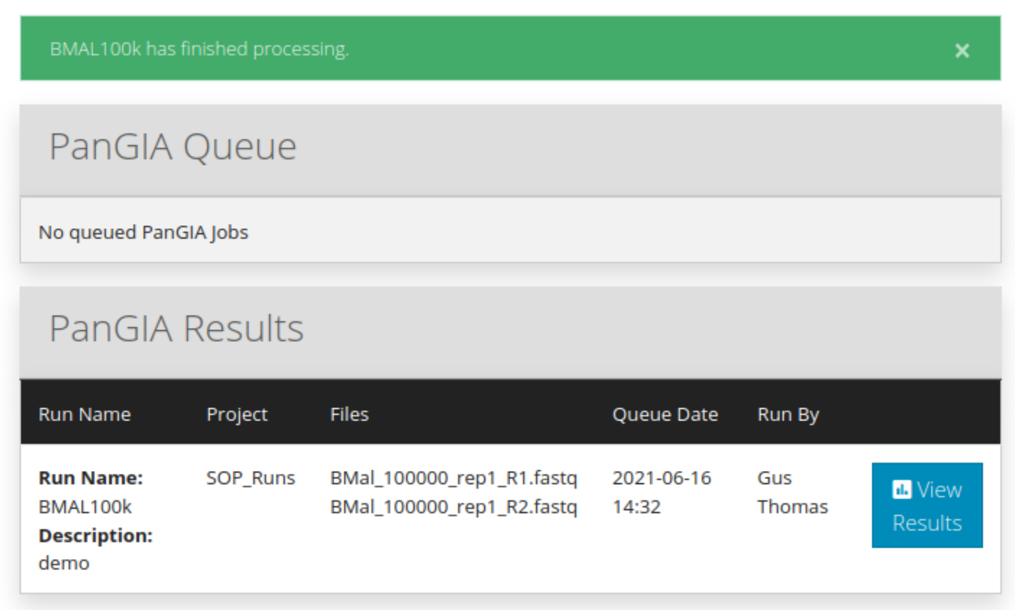
Running Job - BMAL100k

File(s): /home/athomas/Documents/pangia/upload/SOP_Runs/BMal_100000_rep1_R1.fastq /home/athomas/Documents/pangia/upload/SOP_Runs/BMal_100000_rep1_R2.fastq

Description:
demo

```
[00:00:00] Starting PanGIA 1.0.0-RC6.2
[00:00:00] Arguments and dependencies checked:
[00:00:00]     Input reads       : ['/home/athomas/Documents/pangia/upload/SOP_Runs/results/tmp/BMal_
[00:00:00]     Input SAM file    : /home/athomas/Documents/pangia/upload/SOP_Runs/results/tmp/BMal_10
[00:00:00]     Input background  : None
[00:00:00]     Save background   : None
[00:00:00]     Scoring method    : standalone
[00:00:00]     Scoring parameter : 0.5:0.99
[00:00:00]     Database          : ['/home/athomas/Documents/pangia/database/DY_atcc.additions.fa.mmi
[00:00:00]     Abundance         : DEPTH_COV
[00:00:00]     Output path       : /home/athomas/Documents/pangia/upload/SOP_Runs/results/tmp
[00:00:00]     Prefix            : BMal_100000_rep1_R1
[00:00:00]     Mode              : report
[00:00:00]     Specific taxid    : None
[00:00:00]     Threads           : 4
[00:00:00]     First #refs in XA : 30
[00:00:00]     Extra NM in XA    : 1
[00:00:00]     Minimal score     : 0.0
[00:00:00]     Minimal RSNB      : 1
[00:00:00]     Minimal reads     : 3
[00:00:00]     Minimal linear len: 50
[00:00:00]     Minimal genome cov: 0.005
[00:00:00]     Minimal depth (DC): 0.01
[00:00:00]     Minimal RSDCnr    : 0.001
[00:00:00]     Aligner option    : -A1 -B2 -k 40 -m 60 -x sr -p 1 -N 30 --secondary=yes
[00:00:00]     Aligner seed len  : 40
[00:00:00]     Aligner min score : 60
[00:00:00]     Aligner path      : /home/athomas/miniconda3/envs/gui/bin/minimap2
[00:00:00]     Samtools path     : /home/athomas/miniconda3/envs/gui/bin/samtools
[00:00:00] Loading taxonomy information...
[00:00:06] Done.
[00:00:06] Loading pathogen information...
[00:00:06] Done. 2817 pathogens loaded.
[00:00:06] Loading taxonomic uniqueness information...
[00:00:06] Done. 31416 taxonomic uniqueness loaded.
[00:00:06] Loading sizes of genomes...
[00:00:06] Done. 37419 target and 3 host genome(s) loaded.
[00:00:06] Running read-mapping...
[00:00:06] Mapping to /home/athomas/Documents/pangia/database/DY_atcc.additions.fa.mmi...
```

After the job is finished, the Running Job page turns into a Results page. Job completion will be reflected in the PanGIA page of the GUI:



BMAL100k has finished processing.

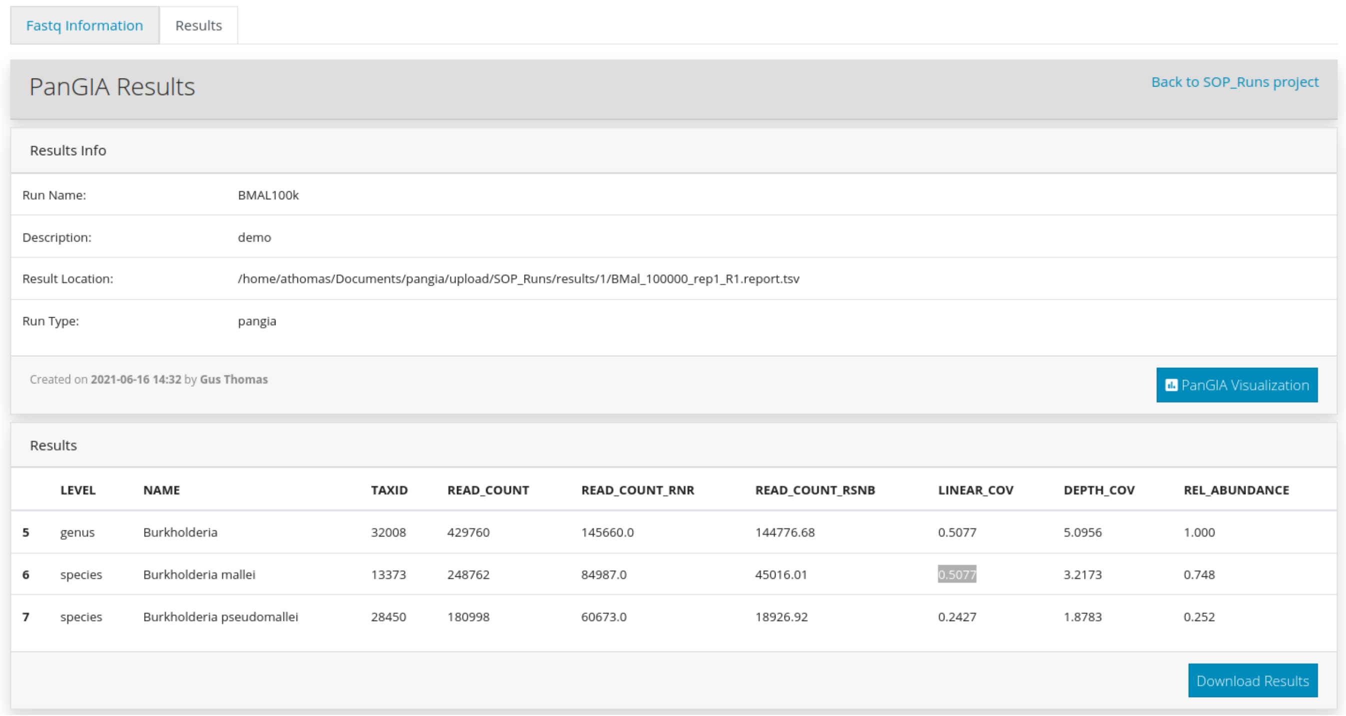
PanGIA Queue

No queued PanGIA Jobs

PanGIA Results

Run Name	Project	Files	Queue Date	Run By	
Run Name: BMAL100k **Description:** demo	SOP_Runs	BMal_100000_rep1_R1.fastq BMal_100000_rep1_R2.fastq	2021-06-16 14:32	Gus Thomas	View Results

Following the 'View Results' link takes us to the Results page, which should look something like this:



PanGIA Results

Results Info

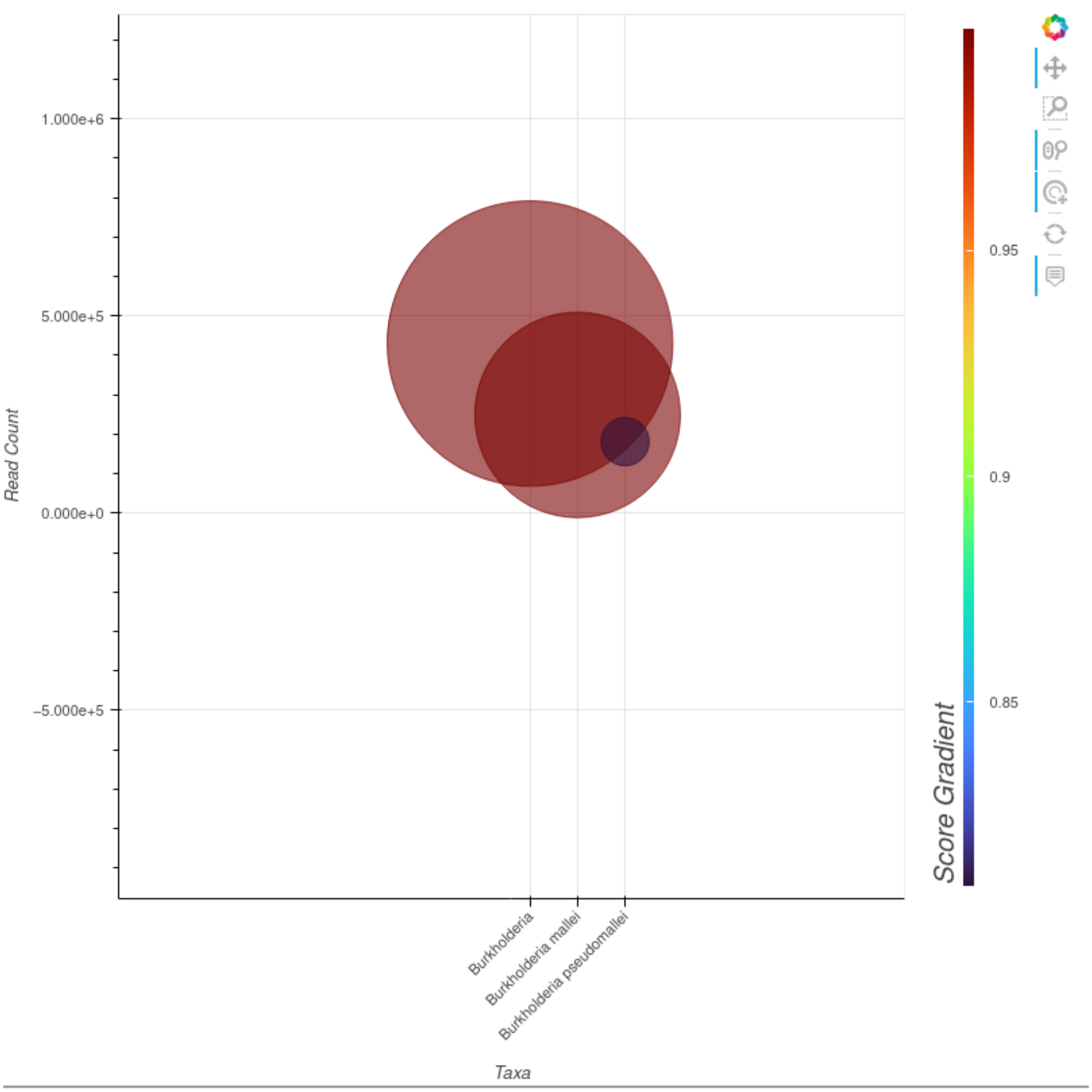
Run Name:	BMAL100k
Description:	demo
Result Location:	/home/athomas/Documents/pangia/upload/SOP_Runs/results/1/BMal_100000_rep1_R1.report.tsv
Run Type:	pangia

Created on **2021-06-16 14:32** by **Gus Thomas**

Results

	LEVEL	NAME	TAXID	READ_COUNT	READ_COUNT_RNR	READ_COUNT_RSNB	LINEAR_COV	DEPTH_COV	REL_ABUNDANCE
5	genus	Burkholderia	32008	429760	145660.0	144776.68	0.5077	5.0956	1.000
6	species	Burkholderia mallei	13373	248762	84987.0	45016.01	0.5077	3.2173	0.748
7	species	Burkholderia pseudomallei	28450	180998	60673.0	18926.92	0.2427	1.8783	0.252

Clicking the PanGIA Visualization button opens the Visualizer in a separate window. More detail concerning planned features are described in **Section 5**, but the scatterplot for the static PanGIA run as described in this section should appear similar to the one depicted here:



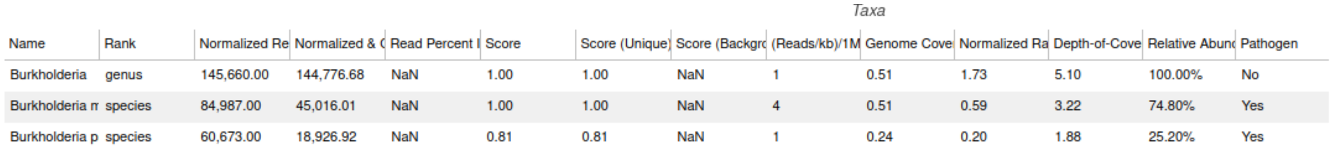
Section 4.3.1.1): Baseline Example Visualizer Usage and Analysis

This section discusses current features of the Visualizer and uses the PanGIA run from **Section 4.3.1** to describe how the Visualizer is used to interpret PanGIA results. Presently, the Visualizer displays hits for taxa between the genus and strain level, and includes a scatterplot and datatable. On the scatterplot, each dot on the plot corresponds to one such taxa and are listed on the x-axis.

1. The scatterplot y-axis indicates 'Raw Read Count' of each hit. This axis is dynamically scaled based on maximum/minimum values present in the sample. In this example, the y-axis covers a very large range of values – between an effective measurement of zero reads and one-million. While this would be too large a scale for comparison of read count between two strain-level hits, it is a suitable scale in this example, where genus *Burkholderia* is compared against species *Burkholderia mallei* and a strain *Burkholderia pseudomallei*. In the image above, the center of the point representing genus *Burkholderia* intersects the y-axis (read count) somewhere between fifty-thousand and five-hundred thousand reads – followed in order of magnitude by species and strain.

2. The scatterplot color-bar 'Score-Gradient' axis on the right indicates hit performance on the selected 'Score' metric – either 'Standalone', 'Background', or 'Combined'. The color-bar is also dynamically scaled; in the above example, strain *Burkholderia pseudomallei* scored lowest among the three hits as indicated by its purplish hue, placing it near the bottom of the color-bar. However, the minimum value for the color-bar is approximately 80%. This result suggests that PanGIA's confidence in correctly identifying this strain is somewhat lower than it's confidence in correctly identifying the species or genus that the strain belongs to, but it still quite high. Dynamically scaled axes make comparison between close-scoring hits easier, given the landscape of hits are restricted to a limited range.

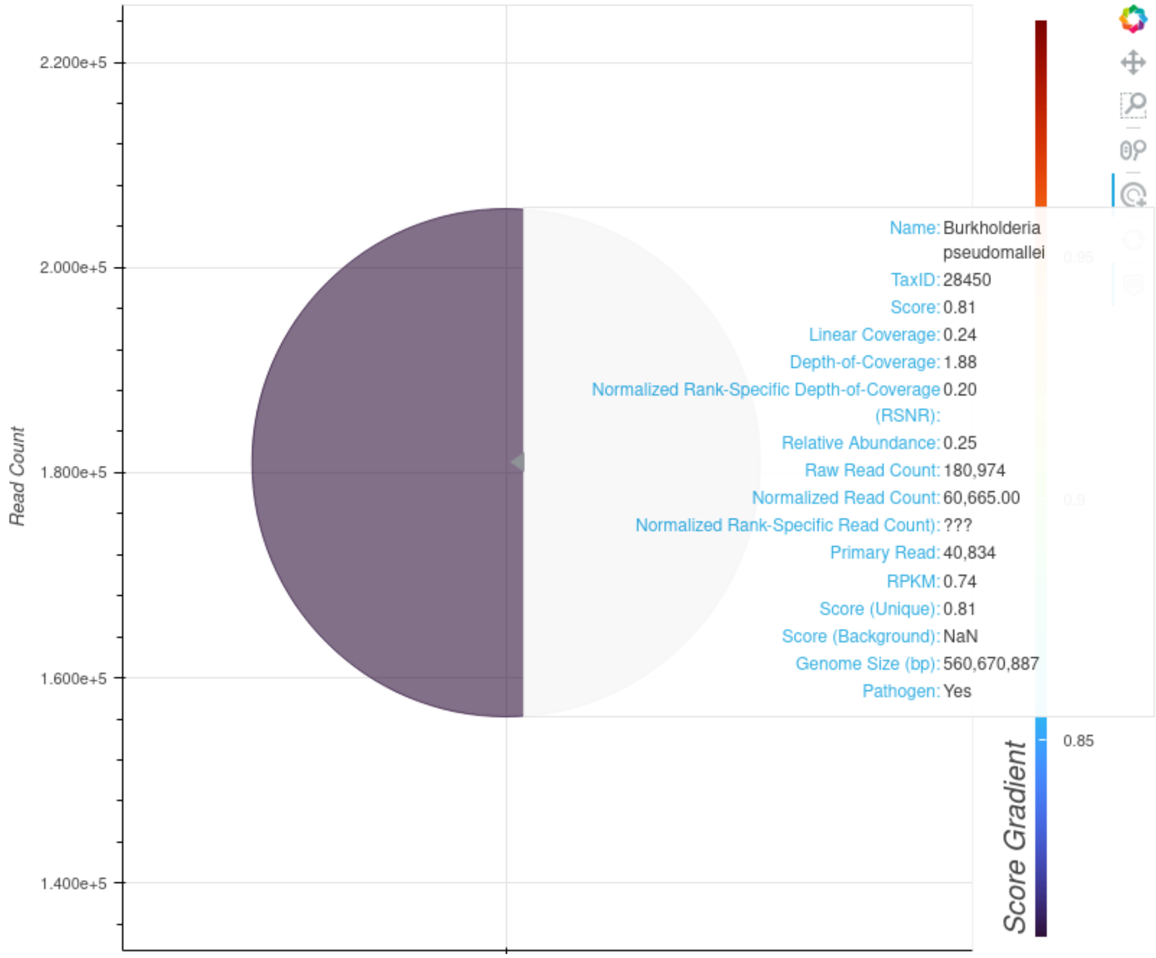
3. The associated data table for this PanGIA run contains three observations – corresponding to the three data points seen in the above scatterplot. A condensed summary of the information in a single row of this table appears as a tool-tip when the mouse is hovered over the center of a dot in the scatterplot.



Taxa

Name	Rank	Normalized Re	Normalized & (	Read Percent I	Score	Score (Unique)	Score (Backgr(	(Reads/kb)/1M	Genome Cove	Normalized Ra	Depth-of-Cove	Relative Abun(	Pathogen
Burkholderia	genus	145,660.00	144,776.68	NaN	1.00	1.00	NaN	1	0.51	1.73	5.10	100.00%	No
Burkholderia m	species	84,987.00	45,016.01	NaN	1.00	1.00	NaN	4	0.51	0.59	3.22	74.80%	Yes
Burkholderia p	species	60,673.00	18,926.92	NaN	0.81	0.81	NaN	1	0.24	0.20	1.88	25.20%	Yes

5. The tool-tip presents numerical results from a given row of the data table. In the below image, *Burkholderia pseudomallei* is shown.



Name: Burkholderia pseudomallei
TaxID: 28450
Score: 0.81
Linear Coverage: 0.24
Depth-of-Coverage: 1.88
Normalized Rank-Specific Depth-of-Coverage (RSNR): 0.20
Relative Abundance: 0.25
Raw Read Count: 180,974
Normalized Read Count: 60,665.00
Normalized Rank-Specific Read Count): ???
Primary Read: 40,834
RPKM: 0.74
Score (Unique): 0.81
Score (Background): NaN
Genome Size (bp): 560,670,887
Pathogen: Yes

Section 5): Planned Visualizer Features

The PanGIA GUI Visualizer is not complete. A depth-scale-down chart for an overview of genome coverage is still in development. We aim to allow the user to customize the dotplot and dynamically adjust the y-axis. We plan to implement a redrawing of the scatterplot each time the user makes an adjustment to any of the filtering tools, such that the taxa listed on the x-axis are removed entirely if one of their values is filtered out.