Intro to IRIDA Pipeline Plugins

Dan Fornika
Micro Binfie Virtual Conference
2020-04-15

Outline

1. Intro to IRIDA
2. How to find and install IRIDA Pipeline Plugins
3. How to develop IRIDA Pipeline Plugins

Intro to IRIDA

IRIDA: **I**ntegrated **R**apid **I**nfectious **D**isease **A**nalysis

An open source, end-to-end platform for public health genomics.

https://www.irida.ca/
https://github.com/phac-nml/irida

IRIDA Features

Data Management

- Projects & Samples
- User accounts with Access Controls

Analysis Pipelines

- Curated Set of Standardized Pipelines
- Additional Pipeline Plugins Available

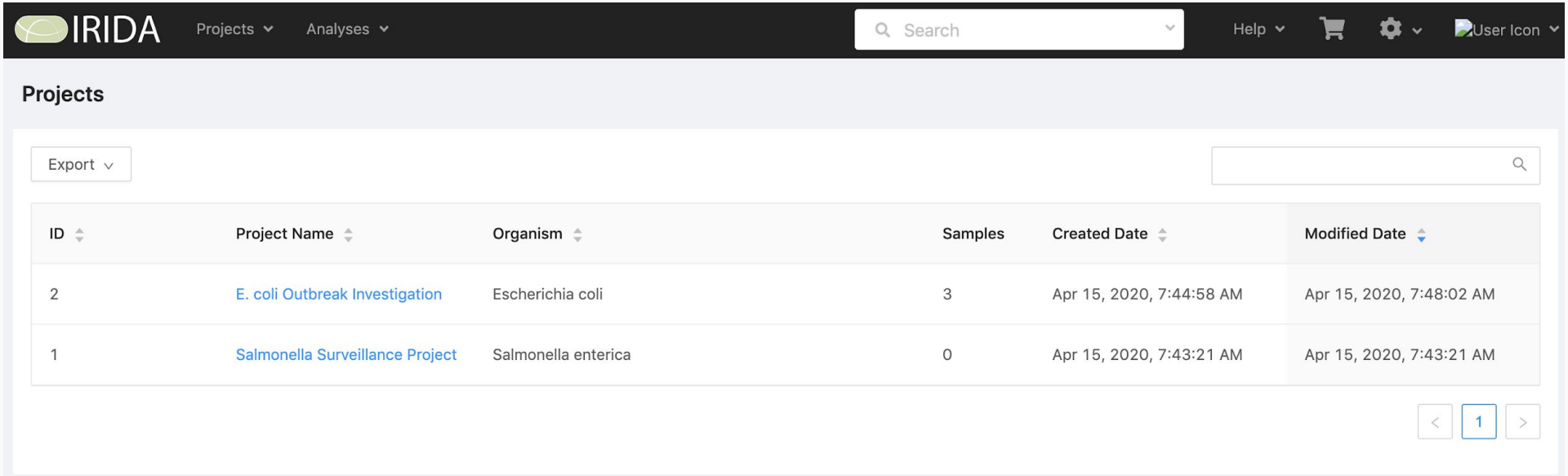
Data Integration

- REST API
- Integrated Metadata Line List

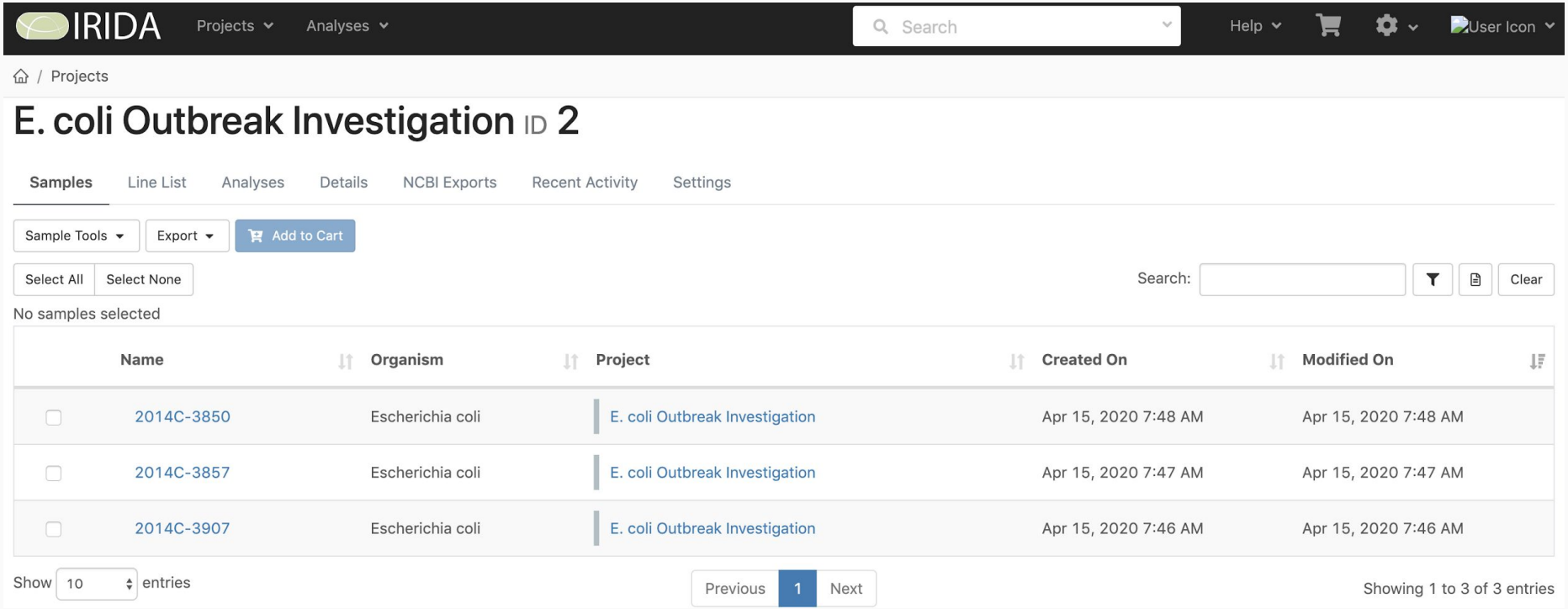
Visualizations

- Interactive Phylogenetic Trees
- Overlay Trees with Sample Metadata

Projects Listing



Samples Listing



Integrated Quality Control

IRIDA Projects Analyses Search Help 3 User Icon

/ Projects / E. coli Outbreak Investigation / Samples / 2014C-3907

2014C-3907_1.fastq

QC Summary | Overrepresented Sequences 0

File Details

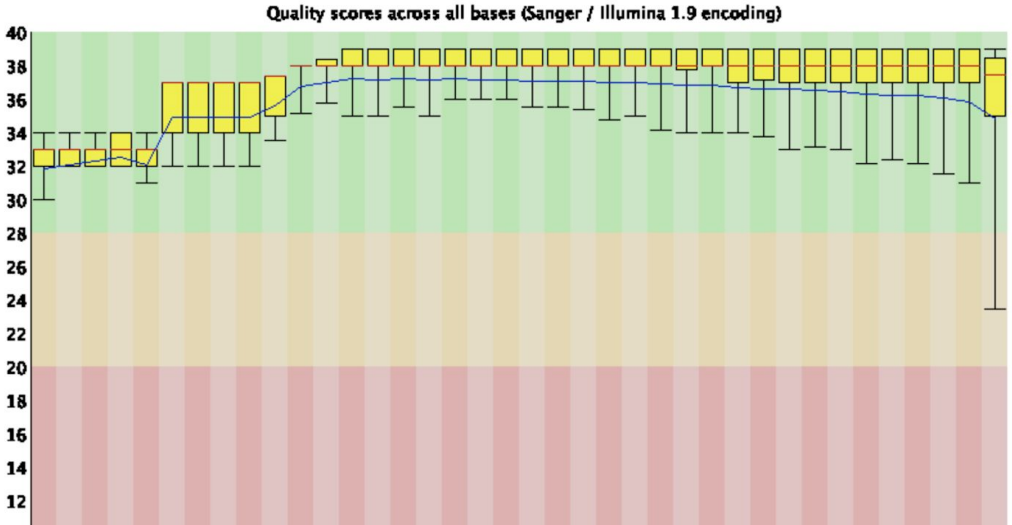
ID	1
Uploaded On	Apr 15, 2020
Encoding	Sanger / Illumina 1.9

Sequence Details

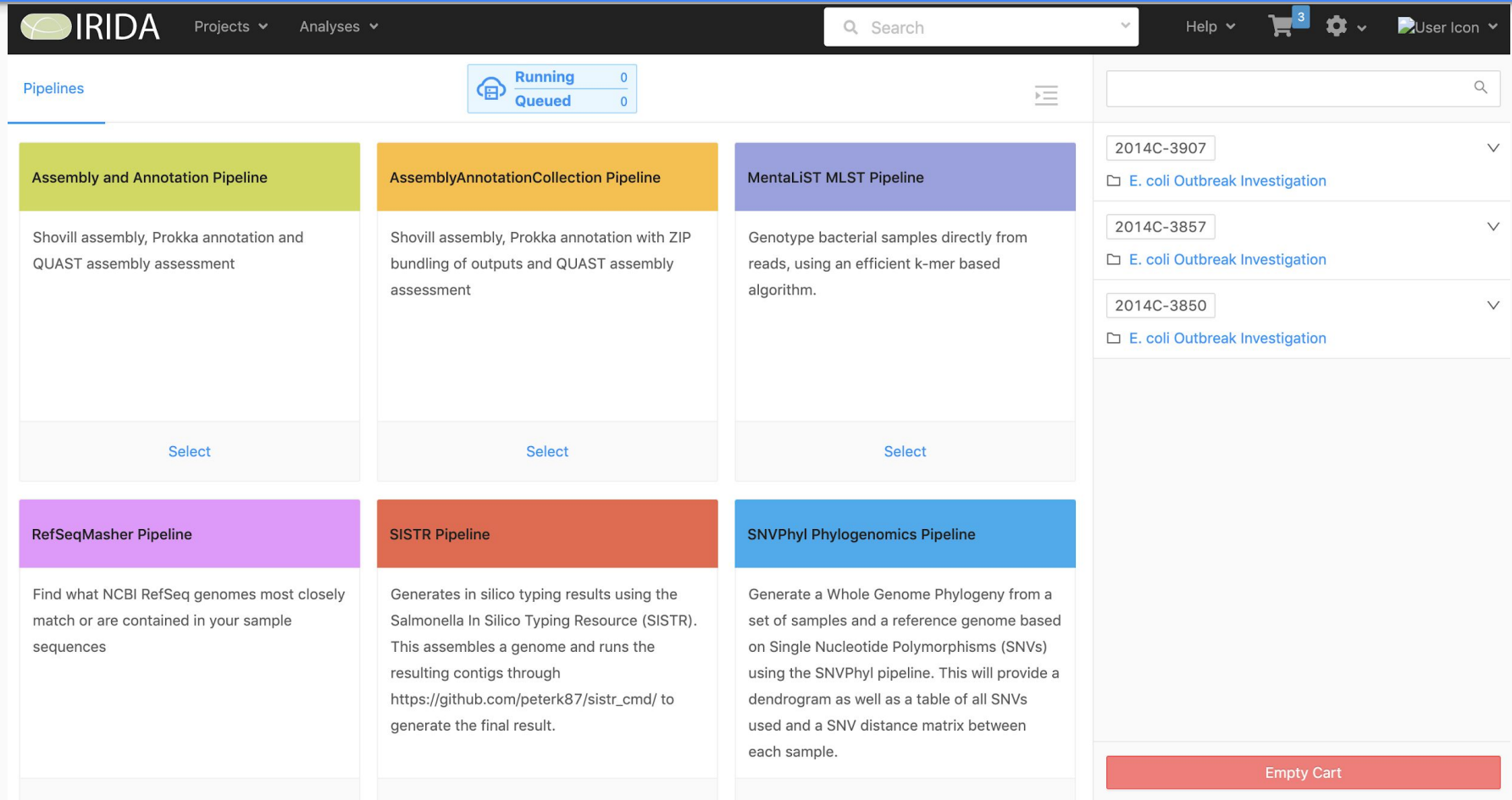
Total Sequences	748141
Total Bases	111377921
Min. Length	35
Max. Length	151
GC Content	50

Quality Charts

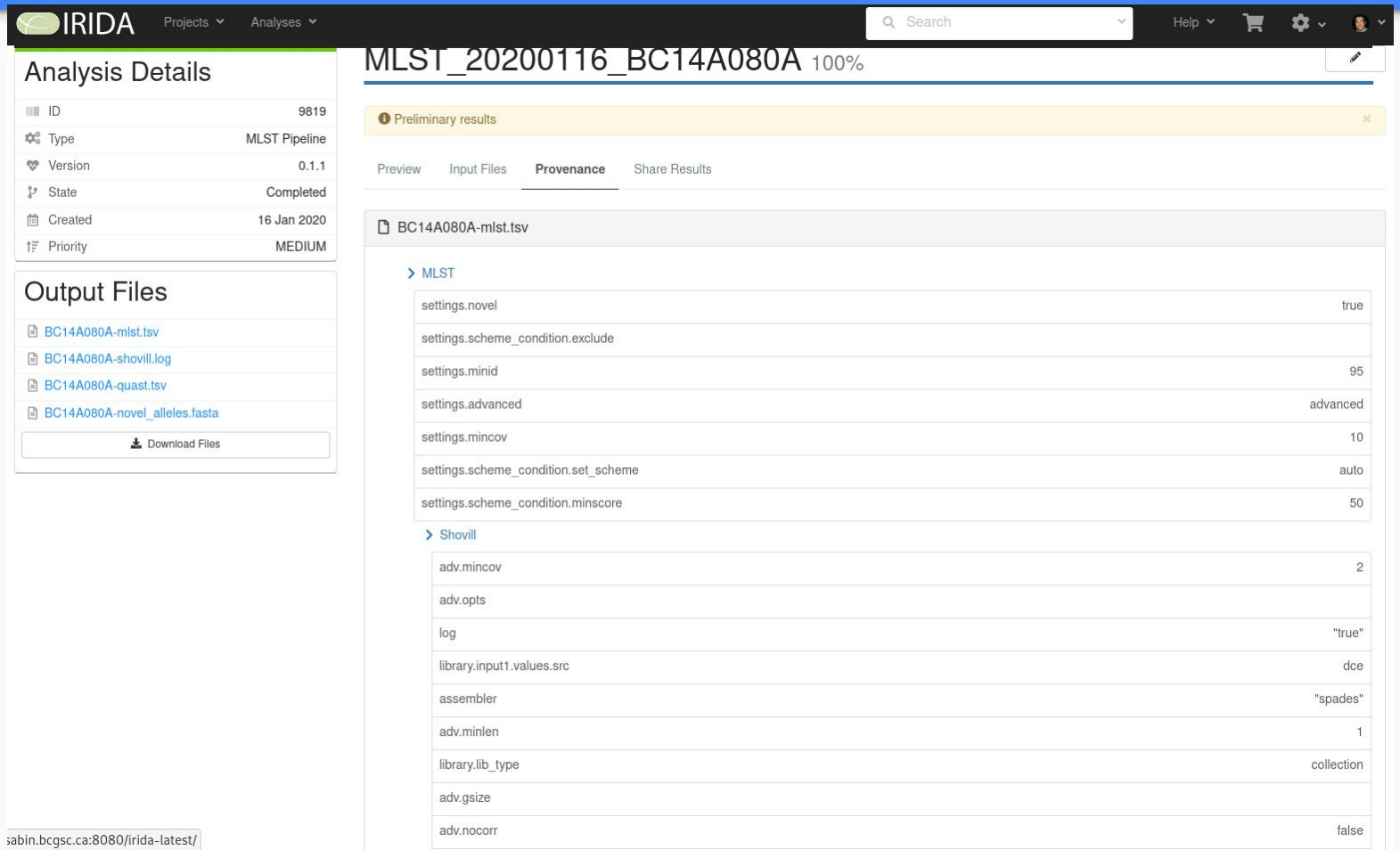
Analysis produced by FastQC (Version 0.11.7)



Analysis Pipeline Selection



Provenance Information



Built-in IRIDA Pipelines

Assembly & Annotation:	Shovill + Prokka
RefSeq Masher:	MASH ‘dist’ and ‘screen’ against RefSeq
SNVPhyl Phylogenomics:	Core SNP Phylogenetics
SISTR:	Salmonella Serotyping
BioHansel:	Salmonella SNV Typing
MentaLiST:	Fast cgMLST

Finding IRIDA Pipeline Plugins

List of available IRIDA Pipeline Plugins:

https://github.com/phac-nml/irida-pipeline-plugins

Installation

- Must be done by system administrator
- Install any necessary Galaxy tools
- Copy single .jar file to plugins directory
- Restart IRIDA

IRIDA Pipeline Plugin Development

Development Environment:

https://irida.corefacility.ca/documentation/developer/setup/

1. Galaxy (Docker): https://github.com/bgruening/docker-galaxy-stable
2. MySQL (Docker)
3. IRIDA source + IDE (IntelliJ IDEA or Eclipse)

Galaxy Tool Development: Planemo: https://github.com/galaxyproject/planemo

Development Environment Setup

Host System

IDE
(Optional)

IRIDA
(Source Code)

"Port mapping"
`-p 3306:3306`

Containers

Galaxy

MySQL®

Simple Docker Container Management

Add the following to your `.bashrc` to make launching docker containers more convenient:

```
launch-galaxy() {
    docker run --name galaxy --rm -d -p 49999:80 \
      -v ~/galaxy-docker/export/:/export/ \
      -e "ENABLE_TTS_INSTALL=True"  \
      bgruening/galaxy-stable:19.01
}
```

Simple Docker Container Management

Add the following to your `.bashrc` to make launching docker containers more convenient:

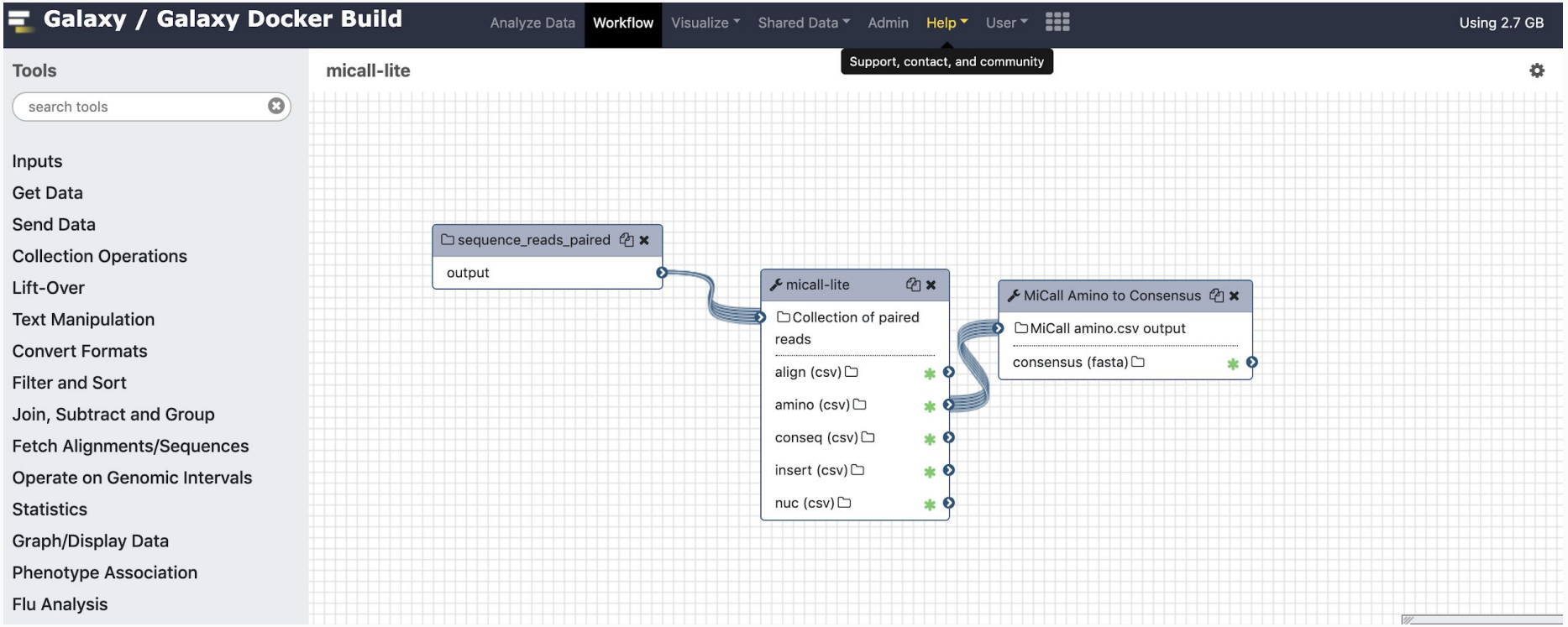
```
launch-mysql() {
docker run --rm -d --name mysql -p 3306:3306 \
   -v ~/mysql-docker/var/lib/mysql:/var/lib/mysql \
   -e "MYSQL_ROOT_PASSWORD=mysql"
   mysql:5.7.4
}
```

Constraints on IRIDA Pipelines

IRIDA Pipelines can take three types of input:

1. .fastq(.gz) sequence files (single-end or double-end)
2. .fasta reference genome (one file per analysis)
3. Entries from Galaxy Tool-Data-Tables

Build Pipeline in Galaxy Docker



Ok, I Have a Galaxy Pipeline. Now What?

IRIDA requires the following files:

`irida_workflow_structure.xml`

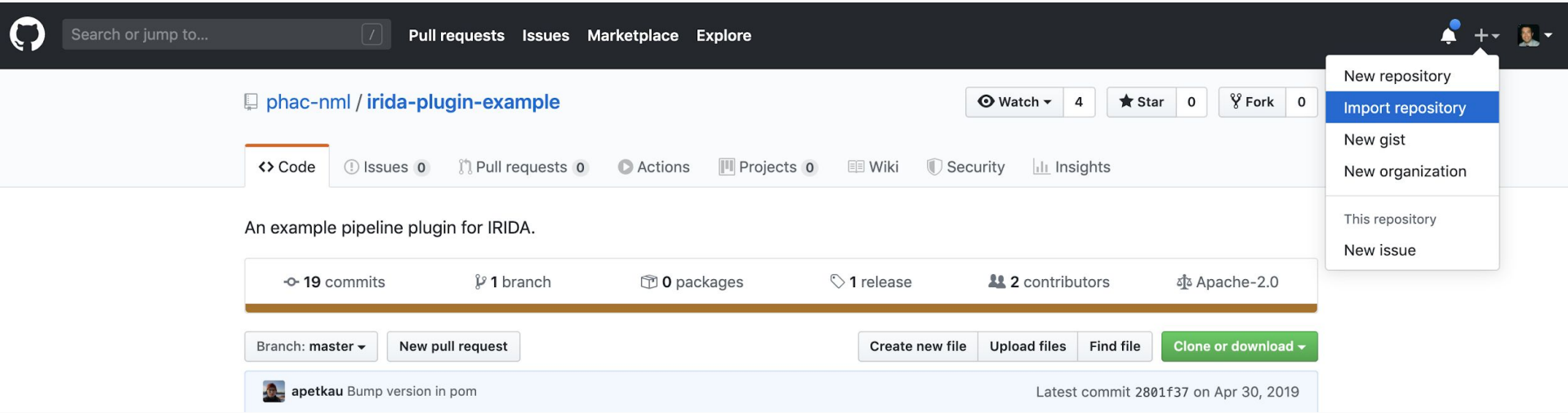
`messages.en.properties`

The ‘irida-wf-ga2xml’ tool is available to automatically generate these files from your galaxy workflow .ga file:

https://github.com/phac-nml/irida-wf-ga2xml

Example Pipeline Plugin

https://github.com/phac-nml/irida-plugin-example



Writing the Plugin Classes

Only 2 java classes necessary for each pipeline plugin:

1. Define plugin details
    a. Unique UUID (https://www.uuidgenerator.net/)
    b. Select a color for the Pipeline Selection Page (google: “colour picker”)
    c. Define ‘AnalysisType’ (simple label for pipeline, eg: “SPECIES_ABUNDANCE”)
2. Metadata Line List Updater (functionality is optional)
    a. Read one (or more) output files from the analysis
    b. Extract specific data element(s) for line-list
    c. Update sample metadata to include new data

Writing to the IRIDA Metadata Line List

Pipeline plugins can be configured to write specific pieces of output data to the IRIDA Metadata Line List

Allows users to accumulate and store small analysis results with samples. Eg:

- MLST sequence type
- Resistance Gene presence/absence

General Guidelines

Include tools to collect QC information

- After running an assembly, run QUAST to assess N50 etc.
- After generating a .bam file, run bamstats

Create metadata fields sparingly.

- Each project has one metadata table.
- Be selective about writing data to the metadata table
- Prefix fields with pipeline name

General Guidelines

Documentation is important!

- The ‘irida-plugin-example’ repository has a great README.md
- Potential users will want to know what the pipeline does before trying it
- Make it clear for sysadmins exactly which tools are required

Short/Simple Pipelines will be more reliable

- It is generally preferable to have a collection of simple pipelines

Thanks!