

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: Integrated Rapid Infectious Disease Analysis

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

Projects

Export ▾

ID ▾	Project Name ▾	Organism ▾	Samples	Created Date ▾	Modified Date ▾
2	E. coli Outbreak Investigation	Escherichia coli	3	Apr 15, 2020, 7:44:58 AM	Apr 15, 2020, 7:48:02 AM
1	Salmonella Surveillance Project	Salmonella enterica	0	Apr 15, 2020, 7:43:21 AM	Apr 15, 2020, 7:43:21 AM

Samples Listing

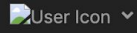


Projects ▾

Analyses ▾

Search

Help ▾



Home / Projects

E. coli Outbreak Investigation ID 2

Samples | [Line List](#) | [Analyses](#) | [Details](#) | [NCBI Exports](#) | [Recent Activity](#) | [Settings](#)

Sample Tools ▾

Export ▾

Add to Cart

Select All

Select None

Search:



Clear

No samples selected

	Name	Organism	Project	Created On	Modified On	
<input type="checkbox"/>	2014C-3850	Escherichia coli	E. coli Outbreak Investigation	Apr 15, 2020 7:48 AM	Apr 15, 2020 7:48 AM	
<input type="checkbox"/>	2014C-3857	Escherichia coli	E. coli Outbreak Investigation	Apr 15, 2020 7:47 AM	Apr 15, 2020 7:47 AM	
<input type="checkbox"/>	2014C-3907	Escherichia coli	E. coli Outbreak Investigation	Apr 15, 2020 7:46 AM	Apr 15, 2020 7:46 AM	

Show entries

Previous

1

Next

Showing 1 to 3 of 3 entries



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

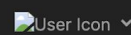


Projects ▾

Analyses ▾

Search

Help ▾



Pipelines

Running 0
 Queued 0



Assembly and Annotation Pipeline

Shovill assembly, Prokka annotation and QAST assembly assessment

Select

AssemblyAnnotationCollection Pipeline

Shovill assembly, Prokka annotation with ZIP bundling of outputs and QAST assembly assessment

Select

MentaLiST MLST Pipeline

Genotype bacterial samples directly from reads, using an efficient k-mer based algorithm.

Select

RefSeqMasher Pipeline

Find what NCBI RefSeq genomes most closely match or are contained in your sample sequences

SISTR Pipeline

Generates in silico typing results using the Salmonella In Silico Typing Resource (SISTR). This assembles a genome and runs the resulting contigs through https://github.com/peterk87/sistr_cmd/ to generate the final result.

SNVPhyl Phylogenomics Pipeline

Generate a Whole Genome Phylogeny from a set of samples and a reference genome based on Single Nucleotide Polymorphisms (SNVs) using the SNVPhyl pipeline. This will provide a dendrogram as well as a table of all SNVs used and a SNV distance matrix between each sample.

2014C-3907



[E. coli Outbreak Investigation](#)

2014C-3857



[E. coli Outbreak Investigation](#)

2014C-3850



[E. coli Outbreak Investigation](#)

Empty Cart

Provenance Information

IRIDA Projects ▾ Analyses ▾ Search Help ▾ Shopping Cart Settings User

Analysis Details

ID	9819
Type	MLST Pipeline
Version	0.1.1
State	Completed
Created	16 Jan 2020
Priority	MEDIUM

Output Files

- [BC14A080A-mlst.tsv](#)
- [BC14A080A-shovill.log](#)
- [BC14A080A-quast.tsv](#)
- [BC14A080A-novel_alleles.fasta](#)

[Download Files](#)

MLST_20200116_BC14A080A 100%

Preliminary results

[Preview](#) [Input Files](#) **[Provenance](#)** [Share Results](#)

BC14A080A-mlst.tsv

> MLST

settings.novel	true
settings.scheme_condition.exclude	
settings.minid	95
settings.advanced	advanced
settings.mincov	10
settings.scheme_condition.set_scheme	auto
settings.scheme_condition.minscore	50

> Shovill

adv.mincov	2
adv.opts	
log	"true"
library.input1.values.src	dce
assembler	"spades"
adv.minlen	1
library.lib_type	collection
adv.gsize	
adv.nocorr	false

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:

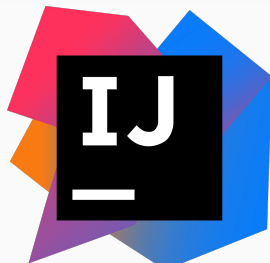
<https://irida.corefacility.ca/documentation/developer/setup/>

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

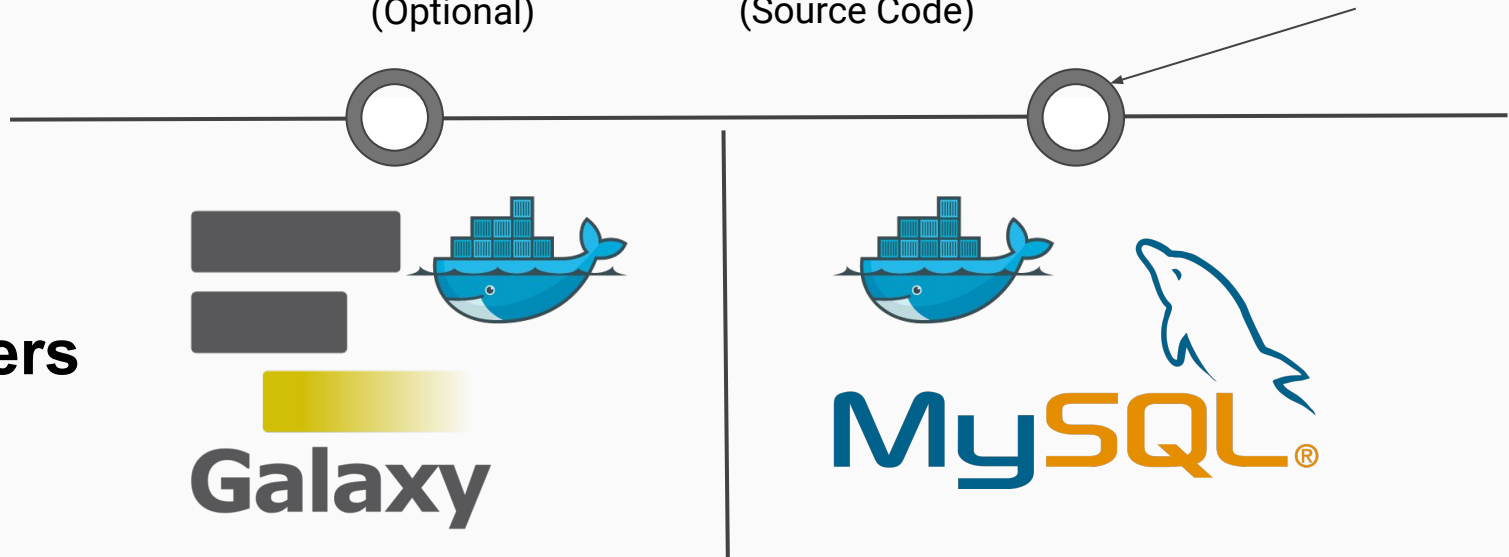
Containers



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

Tools

search tools

Inputs

Get Data

Send Data

Collection Operations

Lift-Over

Text Manipulation

Convert Formats

Filter and Sort

Join, Subtract and Group

Fetch Alignments/Sequences

Operate on Genomic Intervals

Statistics

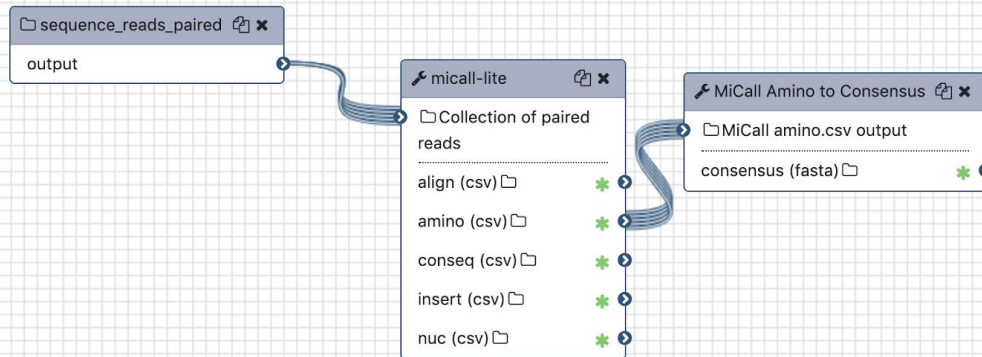
Graph/Display Data

Phenotype Association

Flu Analysis

micall-lite

Support, contact, and community



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>

The screenshot shows the GitHub interface for the repository 'phac-nml / irida-plugin-example'. At the top, there is a navigation bar with the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. On the right side of the navigation bar, there are icons for notifications, a dropdown menu, and a user profile. The repository name 'phac-nml / irida-plugin-example' is displayed in the header, along with 'Watch' (4), 'Star' (0), and 'Fork' (0) buttons. Below the header, there are tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security', and 'Insights'. The main content area contains the text 'An example pipeline plugin for IRIDA.' and a summary bar with statistics: '19 commits', '1 branch', '0 packages', '1 release', '2 contributors', and 'Apache-2.0' license. At the bottom, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A commit history entry is visible at the bottom, showing a commit by 'apetkau' with the message 'Bump version in pom' and the latest commit hash '2801f37' dated 'Apr 30, 2019'. A dropdown menu is open on the right side of the page, showing options: 'New repository', 'Import repository' (highlighted), 'New gist', 'New organization', 'This repository', and 'New issue'.

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!