gga_cloud

Release 0.1

ABiMS

Sep 28, 2021

DEPLOYMENT

Galaxy is an open, web-based platform for accessible, reproducible, and transparent computational biomedical research.

For more information on the Galaxy Project, please visit the https://galaxyproject.org

The Galaxy Genome Annotation (GGA) Project is focused on supporting genome annotation inside Galaxy. It consists of several teams, projects, and tool suites that are working closely together to deliver a comprehensive, scalable and easy to use Genome Annotation experience.

For more information on the Galaxy Project, please visit the https://galaxy-genome-annotation.github.io/

CHAPTER

ONE

GALAXY GENOME ANNOTATION CLOUD INSTALLATION & USAGE

1.1 Installation

1.1.1 Clone

```
git clone https://github.com/abims-sbr/GGA_Cloud.git
```

1.1.2 Dependencies installation

```
ansible-galaxy install -r requirements.yml
```

or to install both separately :

```
ansible-galaxy install role -r requirements.yml
ansible-galaxy install collection -r requirements.yml
```

Need to add `collections_paths = ./collections` in ansible.cfg to use collections.

1.2 Usage

1.2.1 Dry mode

```
ansible-playbook terraform.yml --check
ansible-playbook gga_install.yml --check
ansible-playbook gga_load_data.yml --check
```

1.2.2 Real mode

```
ansible-playbook terraform.yml
ansible-playbook gga_install.yml
ansible-playbook gga_load_data.yml
```

CHAPTER

TWO

GALAXY GENOME ANNOTATION USAGE

2.1 Get data from Galaxy histories

2.2 Build your genome species page

All data loading is done with galaxy, with tools from the Tripal or Chado menu on the left. Prefer the Chado tools as they usually are faster and more reliable.

For a typical genome, you will do something like this:

Chado > Chado organism delete

- Organism: "Homo sapiens"

Chado > Chado organism add

- Genus: Genus
- Species: species ("species rice" if you load multiple variants)
- Common: The famous invulnerable aphid (or gspecies if there is no common name for this species)
- Abbr: G. species ("G. species rice" if you load multiple variants)

Chado > Chado analysis add

- Name: Genus species genome v5.0
- Program: Platanus (if you don't know, write 'Performed by xxx')
- Programversion: v3.1.2c, genome v5.0 (if you don't know, write at least 'genome v5.0')
- Sourcename: Your lab (or the lab who made the assembly)
- Date Executed: The date of creation of the data (if available)

You can populate other if you have more infomation (like source version or execution date)

Chado > Chado analysis add

- Analysis name: Genus species OGS5.0
- Program name: Maker (if you don't know, write 'Performed by xxx')
- Program version: v2.31.8, OGS5.0 (if you don't know, write at least 'OGS5.0')
- Source name: Your lab (or the lab who made the assembly)
- Date Executed: The date of creation of the data (if available)

You can populate other if you have more infomation (like source version or execution date)

Chado > Chado load fasta to load the genome

- Fasta: the genome sequence
- Organism: The organism you created above (if it doesn't appear, wait a few minutes, the list is cached)
- Analysis: The analysis you created above (if it doesn't appear, wait a few minutes, the list is cached)
- Sequence type: contig

Leave the rest unchanged

Chado > Chado load gff

- Gff: the annotation GFF3 file
- Organism: The organism you created above (if it doesn't appear, wait a few minutes, the list is cached)
- Analysis: The analysis you created above (if it doesn't appear, wait a few minutes, the list is cached)
- Landmark type: contig
- Proteins naming method: Regular expression
- Regex protein capture: `^(.*?)-R([A-Z]+)$` (when mRNA ids are the form GSPECIES000001-RA)
- Regex protein: \1P\2 (to convert protein ids to GSPECIES000001-PA)
- Fasta: the protein sequences

Run this only after the following dataset is ready: the output dataset from previous job

Leave the rest unchanged

Tripal > Synchronize features

- Organism: The organism you create above (if it doesn't appear, wait a few minutes, the list is cached)
- Feature types: add 2 items: 'mRNA' and 'polypeptide'

Once this job is finished you should be able to access web pages for loaded mRNAs and polypeptide.

The next step is to load functional annotation.

Tripal > Synchronize an organism

- Organism: The organism you create above (if it doesn't appear, wait a few minutes, the list is cached)

Once this job is finished you should be able to access web pages for loaded mRNAs and polypeptide.

The next step is to load functional annotation.

Tripal > Synchronize an analysis

- Analysis: The genome analysis you created above (if it doesn't appear, wait a few minutes, the list is cached)

Tripal > Synchronize an analysis

- Analysis: The annotation analysis you created above (if it doesn't appear, wait a few minutes, the list is cached)

Chado > Chado analysis add

- Analysis name: Blast2GO on OGS5.0
- Program name: Blast2GO
- Program version: 1.4.4 with 2019.10 database, on OGS5.0
- Source name: Your lab

- Date Executed: The date of creation of the data (if available)

You can populate other if you have more infomation (like source version or execution date)

Chado > Chado load GO annotation

- GO annotation: the gaf file
- Analysis: The analysis you created above (if it doesn't appear, wait a few minutes, the list is cached)
- Organism: The organism you created above (if it doesn't appear, wait a few minutes, the list is cached)
- Query type: polypeptide
- Skip Missing: Yes

Run this only after the following dataset is ready: the output dataset from previous job

You can populate execution date if you know it.

Chado > Chado analysis add

- Analysis name: InterProScan on OGS5.0
- Program name: InterProScan
- Program version: 5.35-74.0, on OGS5.0
- Source name: Your lab
- Date Executed: The date of creation of the data (if available)

Chado > Chado load InterProScan results

- InterProScan results: the xml file
- Analysis: The analysis you created above (if it doesn't appear, wait a few minutes, the list is cached)
- Organism: The organism you created above (if it doesn't appear, wait a few minutes, the list is cached)
- Query type: polypeptide

Run this only after the following dataset is ready: the output dataset from previous job

You can populate execution date if you know it. If you used Blastp:

Chado > Chado analysis add

- Analysis name: Blastp: on OGS5.0 vs NR
- Program name: Blastp
- Program version: 2.6.0, NR 2019-11-29, OGS5.0
- Source name: Your lab
- Date Executed: The date of creation of the data (if available)

If you used Diamond:

Chado > Chado analysis add

- Analysis name: Diamond: on OGS5.0 vs NR
- Program name: Diamond
- Program version: 0.9.25, NR 2019-11-29, OGS5.0
- Source name: Your lab
- Date Executed: The date of creation of the data (if available)

When using either diamond or blastp:

Chado > Chado load Blast results

- Blast results: the xml file
- Analysis: The analysis you created above (if it doesn't appear, wait a few minutes, the list is cached)
- Organism: The organism you created above (if it doesn't appear, wait a few minutes, the list is cached)
- Query type: polypeptide
- Database blasted against: Genbank_protein

Run this only after the following dataset is ready: the output dataset from previous job

You can populate execution date if you know it.

For each of the blast2go, blastp/diamond and interproscan analyses:

Tripal > Synchronize an analysis

- Analysis: The analysis you create above (if it doesn't appear, wait a few minutes, the list is cached)

Tripal > Populate materialized views

Run this only after the following dataset is ready: the output dataset from previous job

Leave the form unchanged

Tripal > Index Tripal data

Run this only after the following dataset is ready: the output dataset from previous job

Leave the form unchanged