

# • Start on the next project

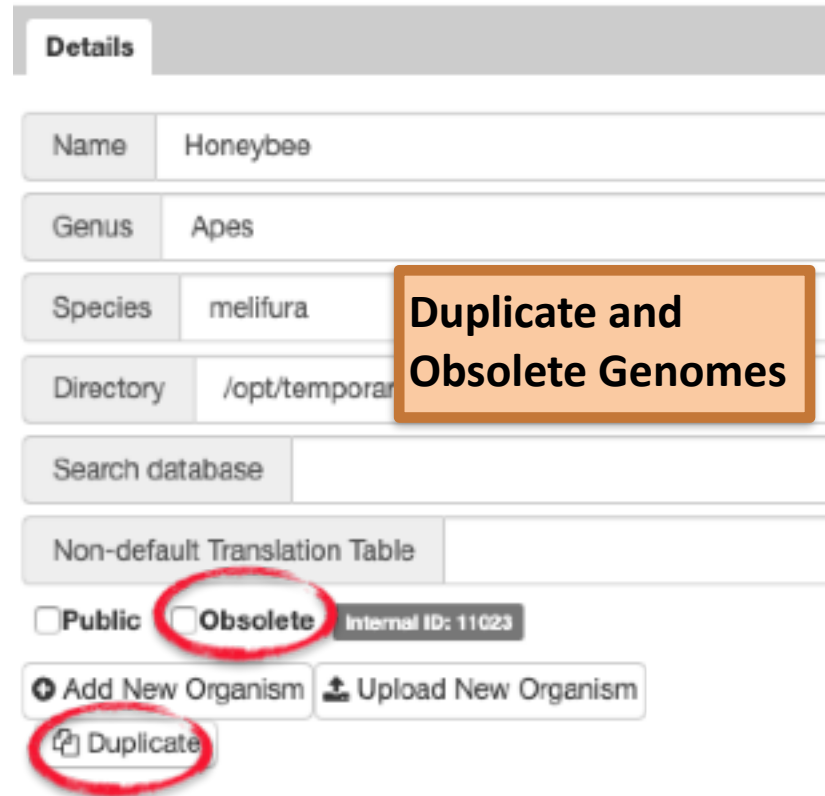
- Plug Apollo into your workflow



The screenshot shows the Apollo genome browser interface. At the top, there's a search bar with 'Honeybee' and 'Group 1.10'. Below the search bar, there's a table with columns 'Name', 'Annotations', and 'Ref Sequences'. The table lists various organisms and their corresponding annotation and reference sequence counts.

Name	Annotations	Ref Sequences
Cat	0	5,500
Celegans	0	7
Chicken	0	1
Cow	1	3,317
Cow2	0	3,317
Dog	0	3,268
Honeybee	80	5,644
Human2 (fail)	0	1
Human-Hg38	6	488
Mouse	0	96

Below the table, there's a 'Details' section with fields for Name, Genus, Species, Directory, Blat database, and buttons for 'Add New Organism' and 'Delete Organism'.



The screenshot shows the 'Details' page for the Honeybee organism. It includes fields for Name, Genus, Species, and Directory. There are also checkboxes for 'Public' and 'Obsolete', and a button for 'Internal ID: 11023'. At the bottom, there are buttons for 'Add New Organism', 'Upload New Organism', and 'Duplicate'.

**Details**

Name: Honeybee

Genus: Apes

Species: melifera

Directory: /opt/tempor

Search database

Non-default Translation Table

☐ Public ☒ **Obsolete** Internal ID: 11023

☒ Add New Organism ☐ Upload New Organism

☒ **Duplicate**

**Duplicate and  
Obsolete Genomes**

# • Plug Apollo into your workflow

[https://toolshed.g2.bx.psu.edu/view/gga/suite\\_apollo/](https://toolshed.g2.bx.psu.edu/view/gga/suite_apollo/)

<https://pypi.org/project/apollo/>

<http://gonramp.wustl.edu/>

@erasche @abretaud

@luke-c-sargent @Yating-L



```
pip install pyarrow
pip install apollo
```



```
$arrow groups create_group university
$arrow users get_users | \
  jq '.[ ] | select(.username | \
contains("@tam.u.edu")) | .username' | \
  xargs -nl arrow users add_to_group
university
```

```
from apollo import ApolloInstance
wa = ApolloInstance('https://fqdn/apollo',
                    'jane.doe@fqdn.edu', 'password')
orgs = wa.organisms.add_organism(
    "Yeast",
    "/path/to/jbrowse/data")
```



{ REST }

Galaxy