Galaxy4Bioinformatics

Développement et intégration d'application sous Galaxy

INSTITUT FRANÇAIS DE BIOINFORMATIQUE

Olivia Doppelt-Azeroual
Fabien Mareuil
Alban Lermine

5 Novembre 2014

API GALAXY

cnrs · IRGI · INRA SCIENCE & IMPACT · Institut Pasteur · SouthGreen bioinformatics platform · cirad · IRD · institutCurie · Abims · CNRS UPMC Station Biologique Roscoff · UPMC SORBONNE UNIVERSITÉS

Présentation API Galaxy

L'API permet l'accès aux différents éléments de Galaxy via des scripts ou des programmes

- Les méthodes HTTP correspondent à des opérations dans Galaxy. Chacune de ces opérations est aussi implémentée sous forme de scripts stand-alone (dans galaxy-dist/scripts/api/) ou fonction python

HTTP Method	common.py Routine	Standalone
GET	display()	display.py
PUT	update()	update.py
POST	submit()	create.py
DELETE	delete()	delete.py

Présentation API Galaxy

L'utilisation de l'API native est loin d'être triviale!

- Peu ou pas de documentation (https://wiki.galaxyproject.org/Learn/API)
- Fonctions prédéfinies limitées
- Développement lourd autour de ces fonctions
  - Appel des fonctions selon la nomenclature:
    - HTTP VERB + noun + extras
    - ex: POST /api/histories { "name": "new" } -> créé un nouvel historique "new"
  - Pour utiliser un élément spécifique (dataset, historique, librairie, dossier, workflows), il faut commencer par lister tous les éléments existants puis sélectionner celui qui nous intéresse et enfin récupérer l'id (qui n'est pas le nom donné à l'élément) afin d'interagir avec celui ci.

```
libs = display(api_key, api_url + 'libraries', return_formatted=False)
library_id = None
for library in libs:
    if library['name'] == data_library:
        library_id = library['id']
```

Présentation API Galaxy

La librairie Bioblend, utiliser l'API Galaxy facilement!

- Librairie Python développée par Enis Afgane (https://github.com/afgane/bioblend)
- Documentation complète d'utilisation (http://bioblend.readthedocs.org/en/latest/)
- Les éléments Galaxy utilisables via Bioblend sont:
    - Galaxy Instance
    - Datasets
    - Genomes
    - Histories
    - Libraries
    - Quotas
    - ToolShed
    - Users
    - Visual
    - Workflows

Présentation API Galaxy

La librairie Bioblend, utiliser l'API Galaxy facilement!

- Récupérer l'id d'une librairie grâce à son nom, il y a une fonction pour ça!

get_libraries(*library_id=None, name=None, deleted=False*) [source]

Get all the libraries or filter for specific one(s) via the provided name or ID. Provide only one argument: `name` or `library_id`, but not both.

If `name` is set and multiple names match the given name, all the libraries matching the argument will be returned.

Return a list of JSON formatted dicts each containing basic information about a library.

Présentation API Galaxy

Bioblend - Element Galaxy Instance

- Point de départ dans la création de vos scripts API
- Une fonction prenant 2 arguments suffit pour déclarer un Objet Galaxy Instance nécessaire afin de lui adresser des commandes via l'API:

```
gi = galaxy.GalaxyInstance(url='http://127.0.0.1:8000', key='your_api_key')
```

Présentation API Galaxy

Bioblend - Interactions avec les Datasets

- 2 fonctions pour interagir avec les datasets:

```
download_dataset(dataset_id, file_path=None, use_default_filename=True, wait_for_completion=False,
maxwait=12000)  [source]
```

```
show_dataset(dataset_id, deleted=False, hda_ldda='hda')
```

- Exemple d'utilisation: télécharger tous les datasets qui contiennent une chaine de caracteres définie dans leur nom

Présentation API Galaxy

Bioblend - Interactions avec les Génomes

- 3 fonctions pour interagir avec les Génomes:

```
get_genomes()  [source]

install_genome(func='download', source=None, dbkey=None, ncbi_name=None, ensembl_dbkey=None,
url_dbkey=None, indexers=None)  [source]

show_genome(id, num=None, chrom=None, low=None, high=None)  [source]
```

- Exemple d'utilisation: installer automatiquement un nouveau génome à partir de son emplacement réseau NCBI, créer les indexes et éditer les fichiers .loc

Présentation API Galaxy

Bioblend - Interactions avec les Historiques

- 23 fonctions pour interagir avec les historiques:

```
create_history(name=None)  [source]
```

```
delete_dataset(history_id, dataset_id)  [source]
```

```
export_history(history_id, gzip=True, include_hidden=False, include_deleted=False, wait=False)  [source]
```

- Exemple d'utilisation: depuis un outil web tierce, créer un nouvel historique et y importer les données générées par l'outil en amont

Présentation API Galaxy

Bioblend - Interactions avec les Librairies

- 15 fonctions pour interagir avec les librairies:

create_library(*name, description=None, synopsis=None*) [source]

set_library_permissions(*library_id, access_in=None, modify_in=None, add_in=None, manage_in=None*) [source]

upload_from_galaxy_filesystem(*library_id, filesystem_paths, folder_id=None, file_type='auto', dbkey='?', link_data_only=None, roles=''*) [source]

- Exemple d'utilisation: Générer automatiquement une librairie contenant le résultat d'une analyse d'un programme tierce et accessible uniquement par les users désignés

Présentation API Galaxy

Bioblend - Interactions avec les Outils

- 7 fonctions pour interagir avec les librairies:

run_tool(*history_id, tool_id, tool_inputs*) [source]

upload_file(*path, history_id, **keywords*) [source]

- Exemple d'utilisation: Uploader un / des fichiers dans un nouvel historique, exécuter un outil sur ces données, puis télécharger le résultat / le rendre disponible dans un nouvel historique

Présentation API Galaxy

Bioblend - Interactions avec le ToolShed

- 3 fonctions pour interagir avec le ToolShed:

get_repositories() [source]

install_repository_revision(*tool_shed_url, name, owner, changeset_revision, install_tool_dependencies=False, install_repository_dependencies=False, tool_panel_section_id=None, new_tool_panel_section_label=None*) [source]

show_repository(*toolShed_id*) [source]

- Exemple d'utilisation: Rechercher une mise à jour sur un outil toolShed, réaliser l'update puis lancer les tests fonctionnels

Présentation API Galaxy

Bioblend - Interactions avec les Utilisateurs

- 7 fonctions pour interagir avec les utilisateurs:

```
create_local_user(username, user_email, password)  [source]
```

```
create_user_apikey(user_id)  [source]
```

- Exemple d'utilisation: Dans le cadre d'une formation Galaxy, créer les comptes pour une liste d'emails, ajouter un quota pour ces utilisateurs, puis renvoyer les informations par mail aux personnes intéressées

Présentation API Galaxy

Bioblend - Interactions avec les Workflows

- 10 fonctions pour interagir avec les workflows:

export_workflow_json(*workflow_id*) [source]

import_workflow_json(*workflow_json*) [source]

run_workflow(*workflow_id, dataset_map=None, params=None, history_id=None, history_name=None, import_inputs_to_history=False, replacement_params=None*) [source]

- Exemple d'utilisation: Importer un workflow depuis un emplacement réseau puis exécuter celui ci et renvoyer les résultats dans un nouvel historique

Trucs et Astuces

Quelques connaissances acquises dans la douleur

Pré-requis :

- authentification externe (LDAP)
- nouvel utilisateur tout frais (vous possédez son mot de passe)

Besoin :

- créer des librairies automatiquement (intègre un pipeline de création de nouvel utilisateur sur la plate-forme)

Problème :

- on ne peut pas créer de librairie pour un utilisateur qui ne s'est jamais connecté

Solution proposée :

- établie un premier contact avec lynx

```
cmd="lynx -dump -auth=\"%s:%s\" %s > /dev/null" % (user,user_password,url)
```

Trucs et Astuces

Quelques connaissances acquises dans la douleur

Problème :

- Faire cohabiter l'authentification externe (LDAP) avec l'utilisation de l'API
- Votre apache ne permettra le passage de requêtes que si vous avez rentré votre login mot de passe, les requêtes via l'api en ligne de commande n'intègrent pas le login, mot de passe (enfin il ne vaut mieux pas)

Solution proposée :

- Démarrez 2 serveurs Galaxy :
  - le premier configuré normalement pour l'utilisation via browser de l'instance galaxy avec une authentification externe (LDAP)
  - le deuxième ne permettant ni la création d'utilisateur ni l'authentification anonyme, mais sans authentification externe (LDAP) (ouvert à tous)
- Vous pourrez alors utiliser le serveur "api" pour votre utilisation de l'api en ligne de commande

Quelques connaissances acquises dans la douleur

Problème :

- La méthode actuelle de download_dataset ne fonctionnera pas si vous êtes en https ou si vous avez choisi l'option require_login = True dans la configuration de galaxy

Solution :

- Une nouvelle version de la méthode download_dataset est en cours de réflexion dans l'équipe qui développe bioblend

Quelques connaissances acquises dans la douleur

Problème :

- Quand j'ajoute un nouvel utilisateur en accès sur une librairie, il devient le seul à y accéder

Solution :

- Lorsque l'on souhaite ajouter un utilisateur en accès sur une librairie, il faut fournir la liste complète des utilisateurs (ceux qui y avait déjà accès + les nouveaux)

Quelques connaissances acquises dans la douleur

Problème :

- Lorsque l'on *upload* des données dans une librairie depuis un *path system*, si le répertoire défini contient trop de sous répertoires, *l'upload* se termine en erreur et les données ne sont pas accessibles (problème de pool d'accès à la base de données)

Solution :

- Créer un script qui va naviguer dans les différents sous répertoires (récursif) du *path system*, créer le dossier équivalent dans la librairie Galaxy, puis créer autant de jobs d'*upload* que de fichiers présents dans le sous répertoire