PLASMA

PLATEFORME D'E-LEARNING POUR L'ANALYSE DE DONNÉES SCIENTIFIQUES MASSIVES

POWERED BY jupyter

Plasma : plateforme d'e-learning pour l'analyse interactive de données

Jérémy Tuloup Claire Vandiedonck Sandrine Caburet Pierre Poulain

QuantStack Scientific Computing

jres MARSEILLE 2021 2022

Université Paris Cité

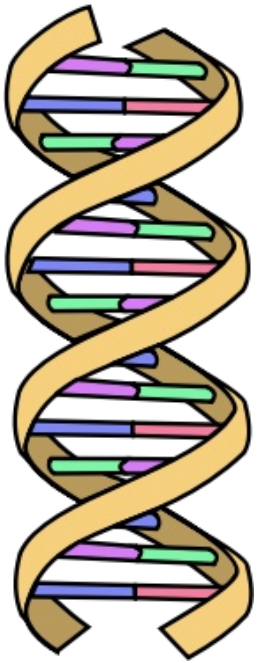
Sauf mention contraire, ce contenu est mis à disposition selon les termes de la licence
Creative Commons Attribution - Partage dans les mêmes conditions 4.0 International (CC BY-SA 4.0)

Besoins initiaux (2016-2018)

Formation « par la recherche » à l'université (L → M)

Analyse de données en bioinformatique génomique

- Programmation (Bash, R, Python)
- Utilisation de logiciels métiers (analyse RNA-seq)



Pierre Poulain CC BY-SA

Cahier des charges (2018)

1. Puissance de calcul
   (de 1 à 8 coeurs, de 2 à 32 Go RAM / étudiant)

2. Accès en dehors du réseau de l'université

3. Interface « intuitive »
   (éviter le terminal, préférer une interface web et des *notebooks*)

Les *notebooks*

GC_content.ipynb

Markdown — Python 3 (ipykernel)

Calcul du pourcentage de GC (*GC content*)

Le %GC est la proportion de bases G et C dans une séquence donnée :

$$\%GC = \frac{nbG+nbC}{nbA+nbT+nbC+nbG} \times 100$$

Soit la séquence :

```
ACGCGATTAGCTAGCCGG
```

[1]:
```
nb_A = 4
nb_T = 3
nb_C = 5
nb_G = 6
```

[2]:
```
GC = (nb_G + nb_C)/(nb_A + nb_T + nb_G + nb_C) * 100
```

[3]:
```
print(GC)
```
```
61.111111111111114
```

[4]:
```
print(f"Le %GC vaut : {GC}")
```
```
Le %GC vaut : 61.111111111111114
```

[5]:
```
print(f"Le %GC vaut : {GC:.1f}")
```
```
Le %GC vaut : 61.1
```

%GC en PCR

Polymerase chain reaction : technique d'amplification de séquences d'ADN

Launcher — volcano_plot.ipynb

Markdown — Python 3 (ipykernel)

Représentation *volcano plot*

Nous utilisons les méthodes *NumPy* `np.log2()` et `np.log10()` pour transformer les valeurs de *fold change* et de p-value.

[8]:
```
exp_df["log fold change"] = np.log2(exp_df["fold change"])
exp_df["log p-value"] = -np.log10(exp_df["p-value"])
```

On affiche encore les premières du jeu de données :

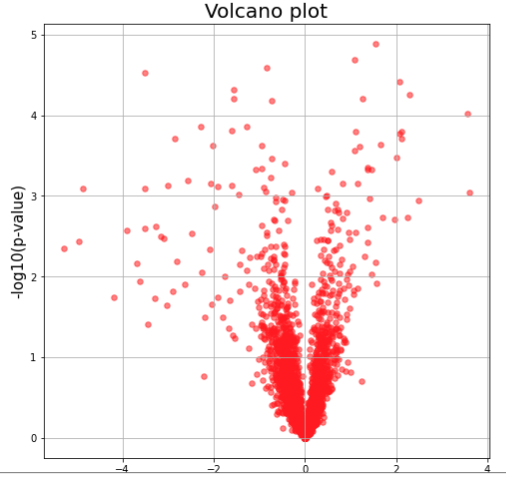
[9]:
```
exp_df.head()
```

[9]:

accession	abundance WT	abundance FRDA	p-value	fold change	log fold change	log p-value
P01876	4.650254e+05	5.488983e+06	0.000095	11.803619	3.561157	4.021460
P28289	3.171518e+04	9.224471e+04	0.000013	2.908535	1.540293	4.889288
P01871	1.596695e+06	5.501002e+04	0.000808	0.034452	-4.859250	3.092484
Q9H0E2	6.336450e+04	1.354001e+05	0.000021	2.136844	1.095482	4.681740
P01857	8.030646e+05	1.104635e+05	0.000196	0.137552	-2.861946	3.708813

Représentons maintenant $-\log_{10}(\text{p-value})$ en fonction de $\log_2(\text{fold change})$:

[10]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot(1,1,1)
ax.set_xlabel("log2(fold change)", fontsize = 15)
ax.set_ylabel("-log10(p-value)", fontsize = 15)
ax.set_title("Volcano plot", fontsize = 20)
ax.scatter(exp_df["log fold change"], exp_df["log p-value"],
           s=30, alpha=0.5, c="red")
ax.grid()
```



Cahier des charges (2018)

1. Puissance de calcul
   (de 1 à 8 coeurs, de 2 à 32 Go RAM / étudiant)

2. Accès en dehors du réseau de l'université

3. Interface « intuitive »
   (éviter le terminal, préférer une interface web et des *notebooks*)

4. Solution polyvalente pour différents types d'enseignements

5. Déploiement et administration « simples »
   (pas de cluster, pas K8s)

Solution technique : écosystème Jupyter

- **Jupyter Hub** : portail de connexion / authentification
- **Jupyter Server** : gestion des interfaces d'analyse (JupyterLab, RStudio)
- **JupyterLab** : interface d'analyse intégrée (*notebooks*, explorateur de fichiers, terminal, éditeur de fichiers...)

+ **environnements « étanches » et configurables** (langages, bibliothèques, logiciels, données, scénarios)

Financement du projet (2018-2021)

200 k€

Serveurs + prestation développement + communication

Implémentation et développement (2020)

Configuration matérielle :

2 x Dell R840

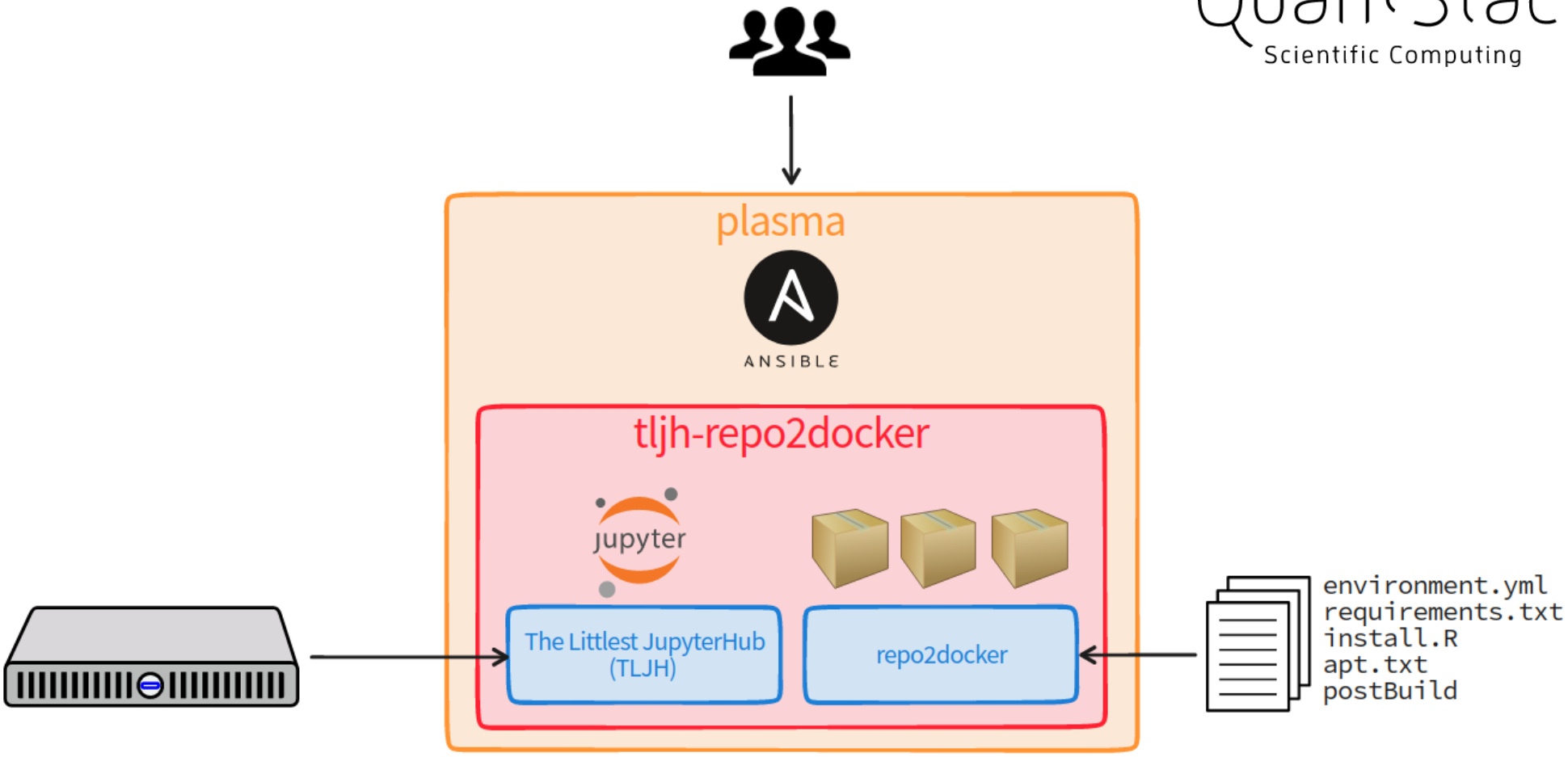
- 80 coeurs
- 768 Go RAM
- 30 To stockage

Prérequis techniques :

- Ubuntu 18.04 / 20.04
- Accès SSH et HTTPS

Implémentation et développement (2020)

QuantStack
Scientific Computing



Implémentation et développement (2020)

QuantStack
Scientific Computing

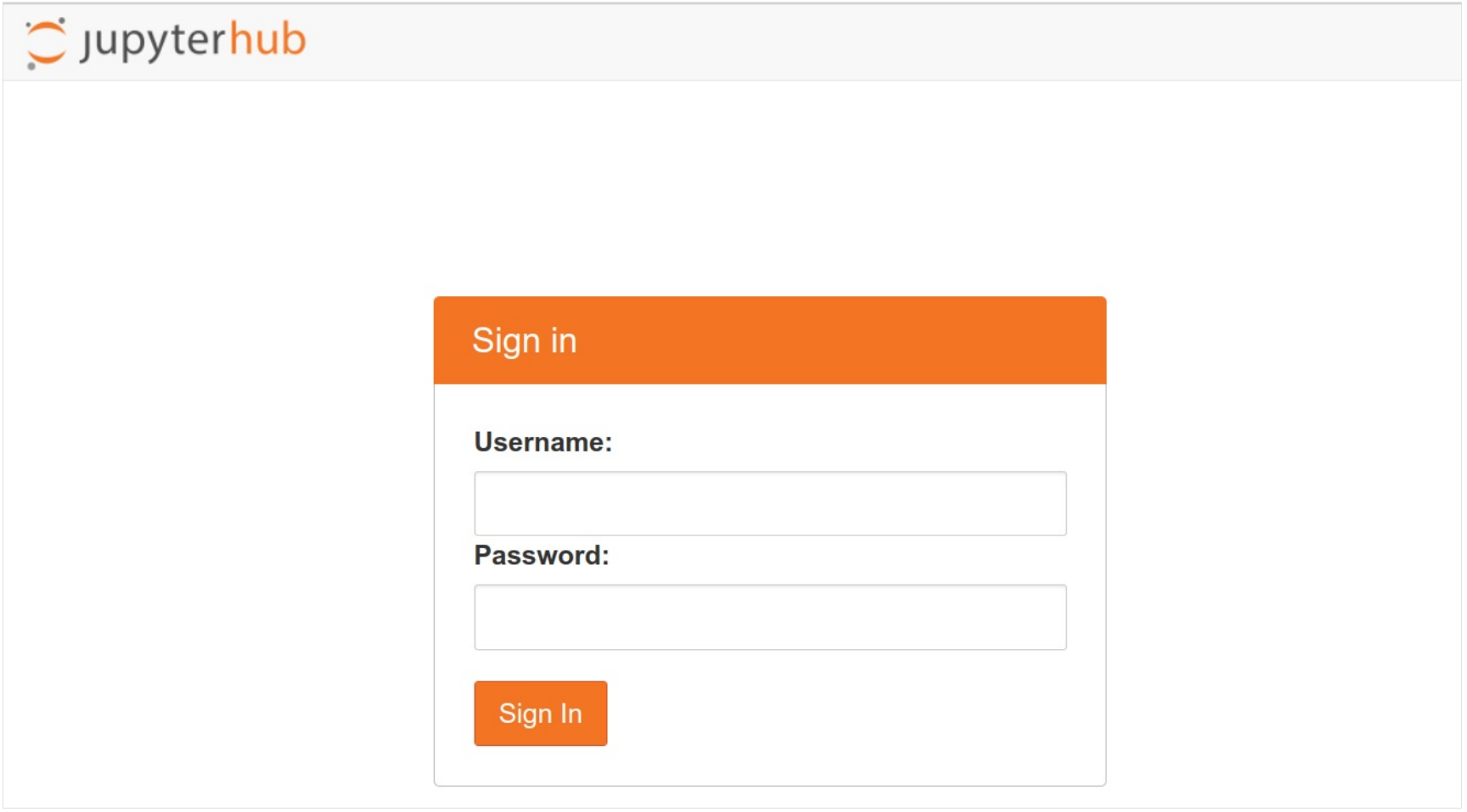
tljh-repo2docker

- The Littlest JupyterHub (JupyterHub) + repo2docker
- https://github.com/plasmabio/tljh-repo2docker

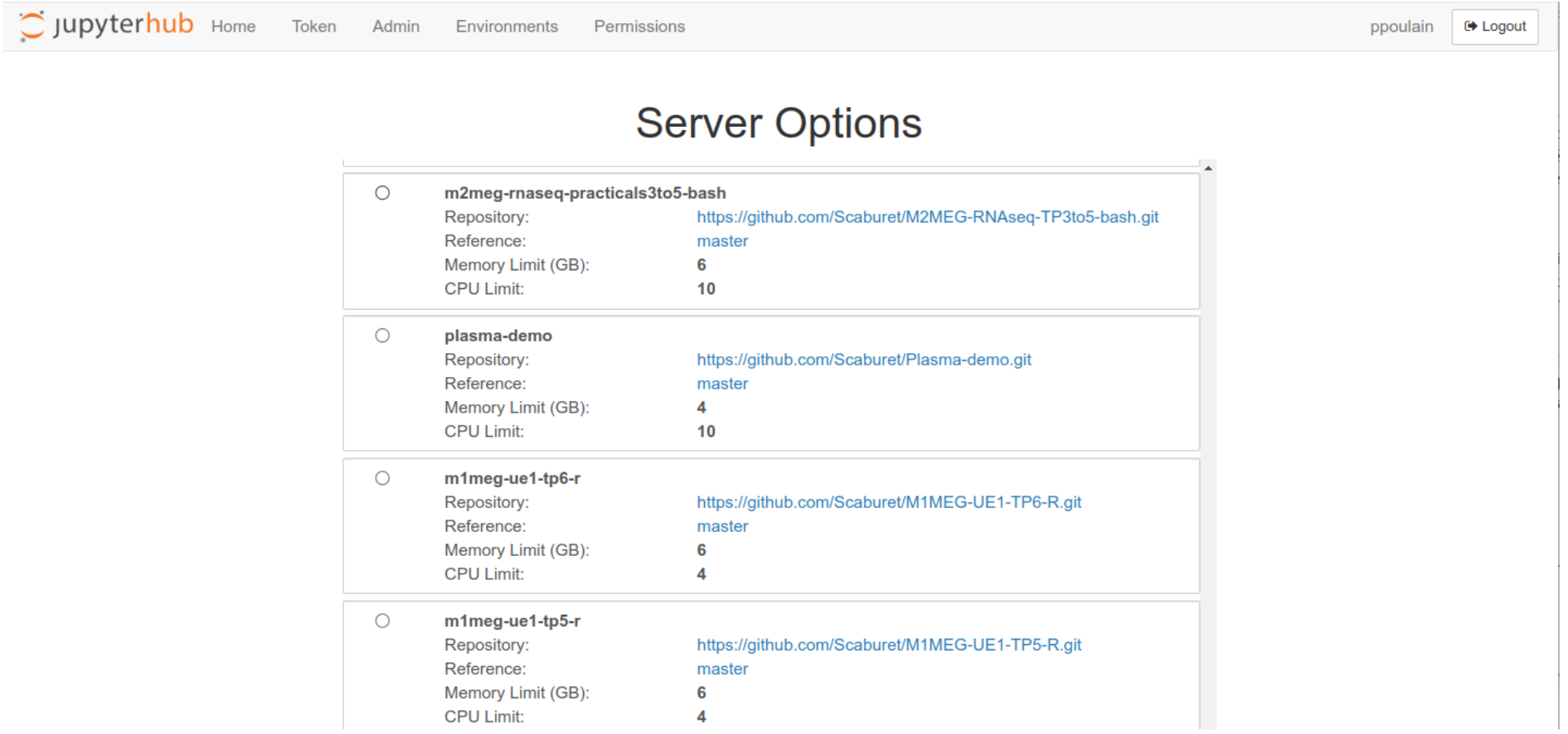
plasma

- *playbooks* Ansible pour le déploiement et la configuration
- https://github.com/plasmabio/plasma

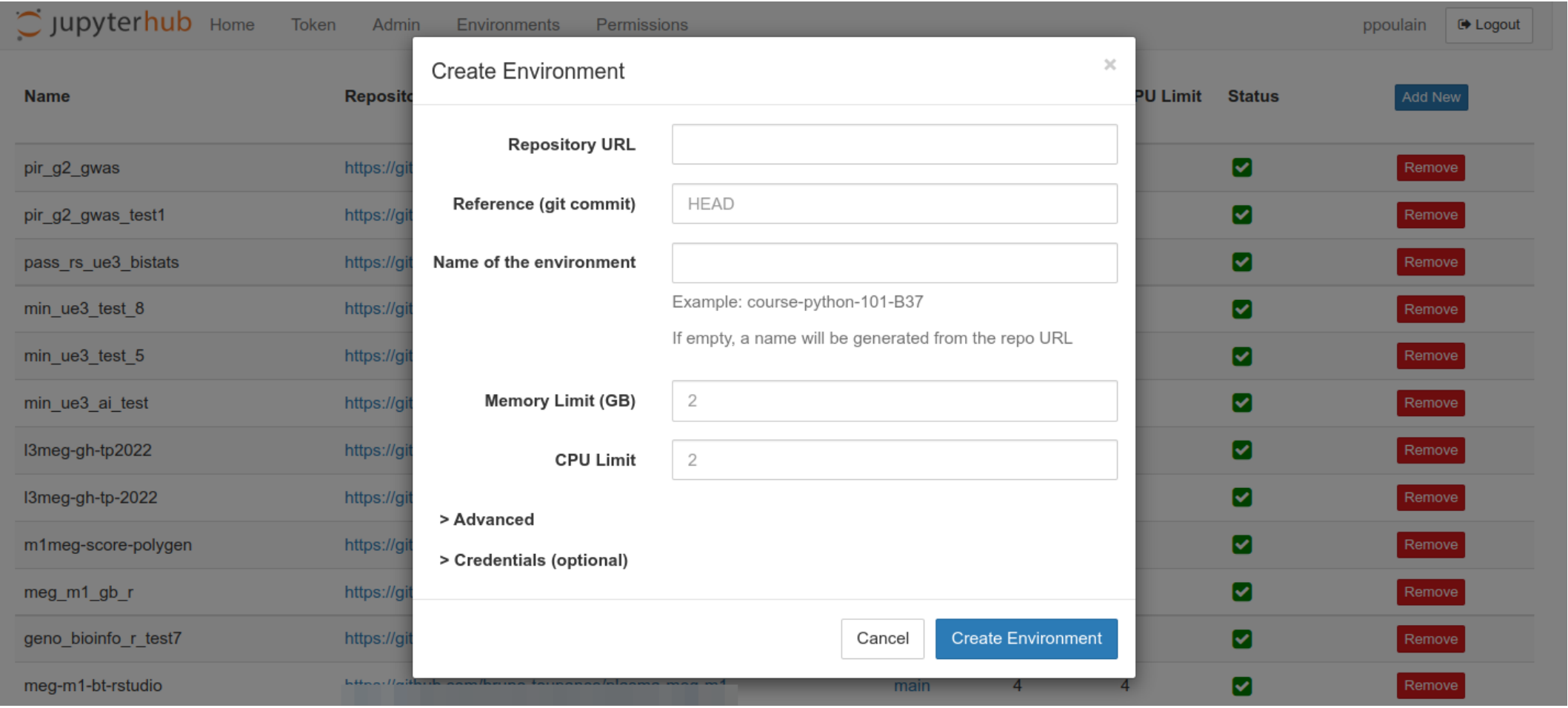
Interface de connexion



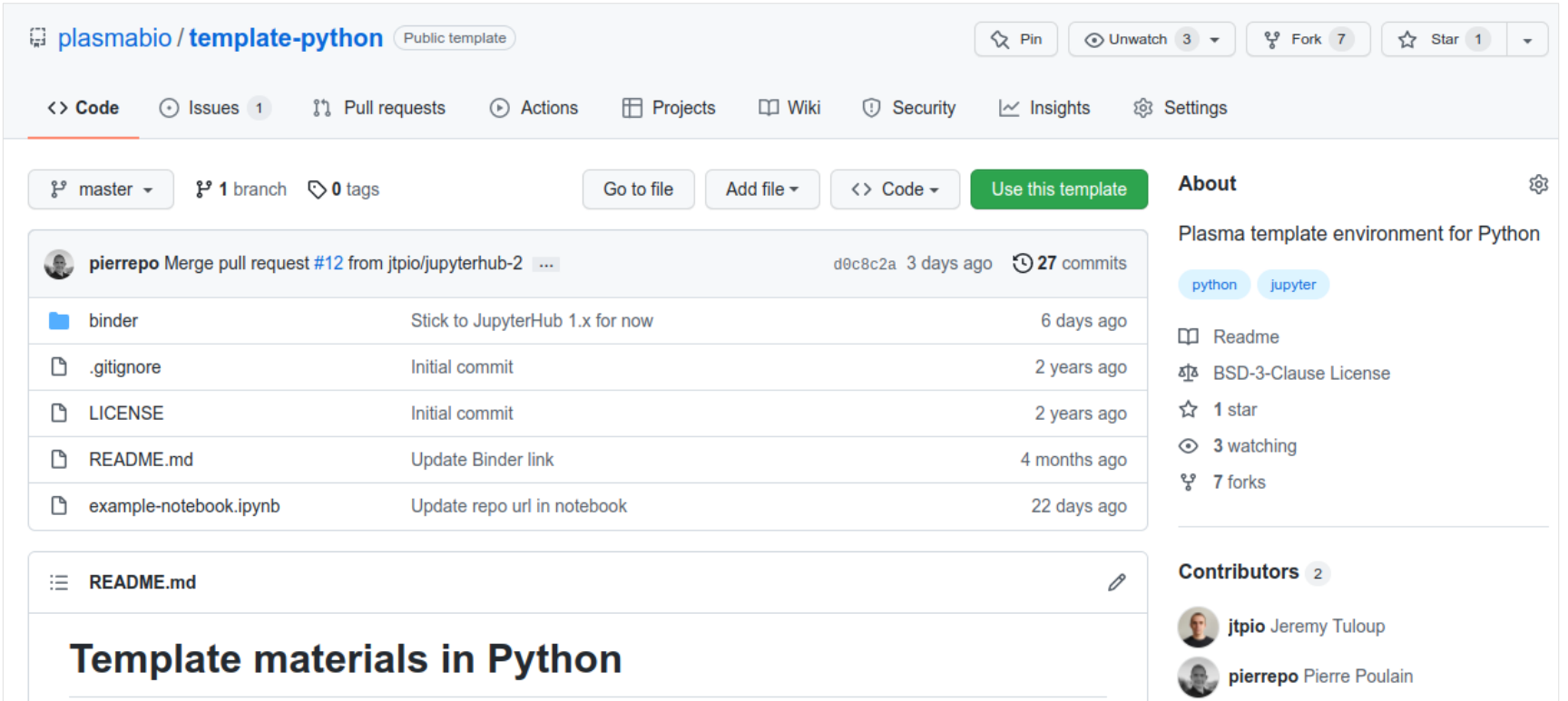
Choix des environnements



Construction d’un environnement

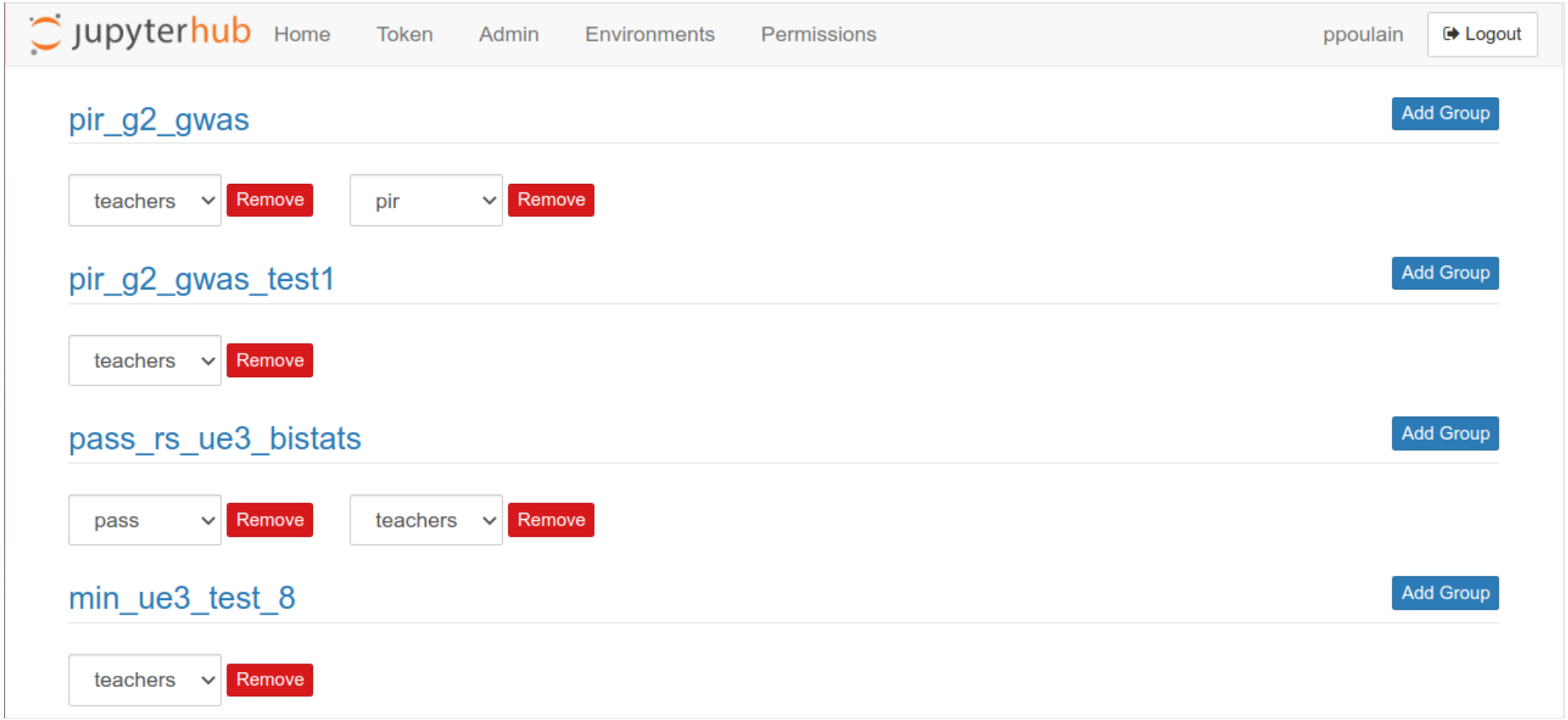


Modèles d’environnements

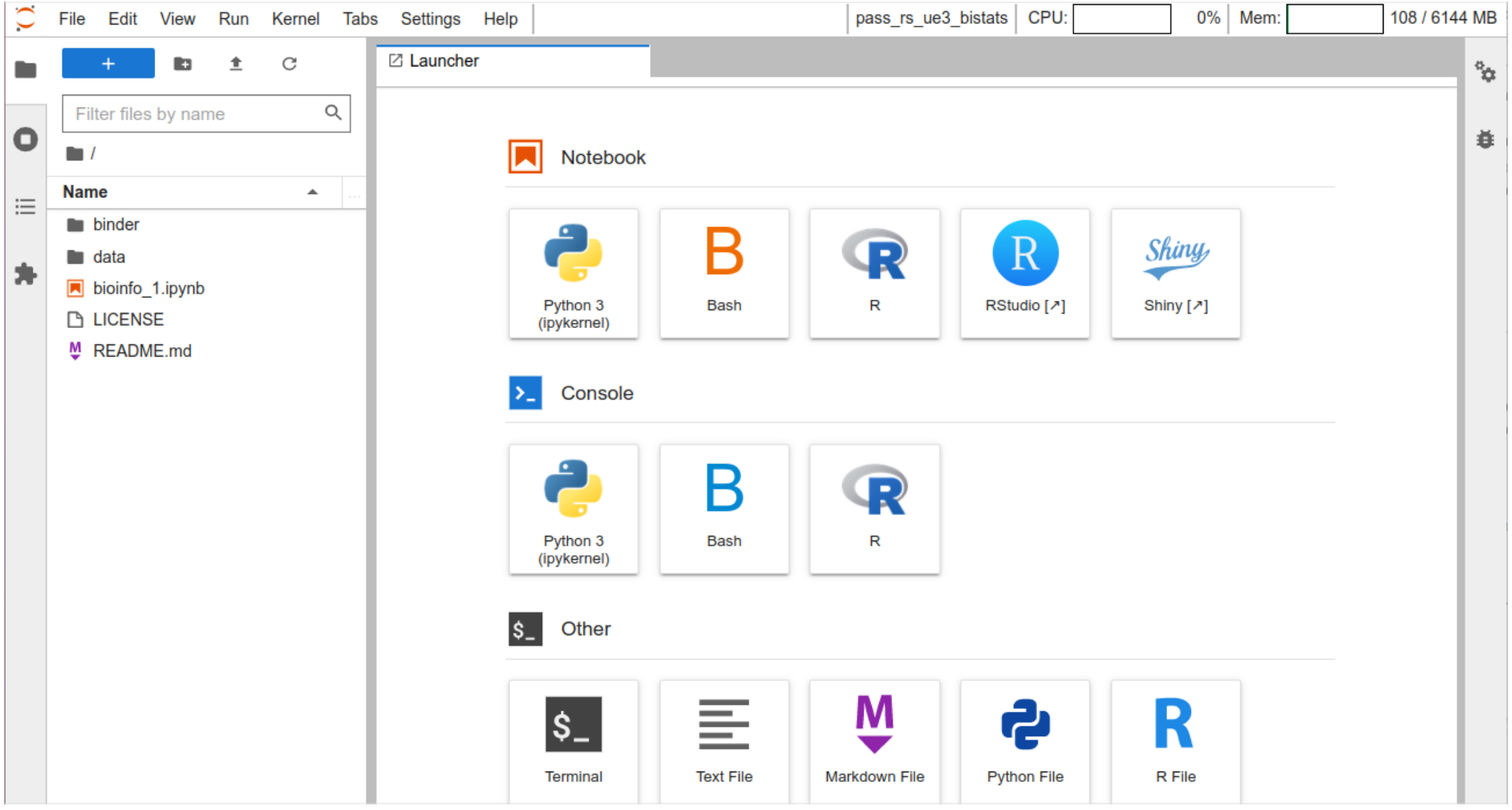


https://github.com/plasmabio/template-python
https://github.com/plasmabio/template-r
https://github.com/plasmabio/template-bash

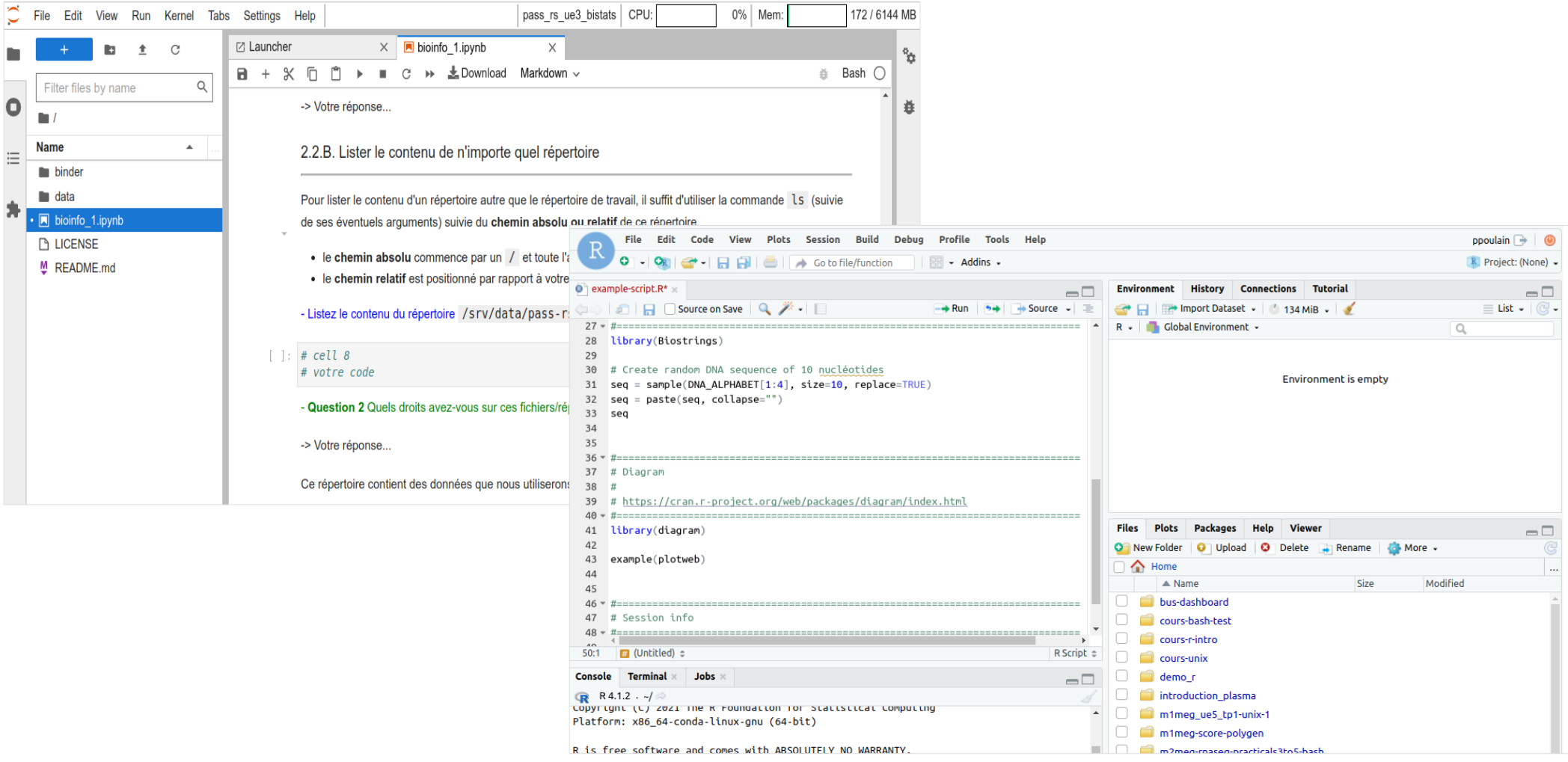
Définition des permissions



Interface JupyterLab



Interfaces d’analyse JupyterLab / RStudio



Retours utilisateurs (étudiants)

« Très pratique, facilite l'**accès à notre travail partout**, vraiment utile. »

« Très agréable et a permis de suivre les cours **sans avoir à m'embêter à tout installer** sur ma machine. »

« J'ai beaucoup aimé les notebooks, car ils permettent de bien **organiser le travail** et de revenir dessus quand on veut. ».

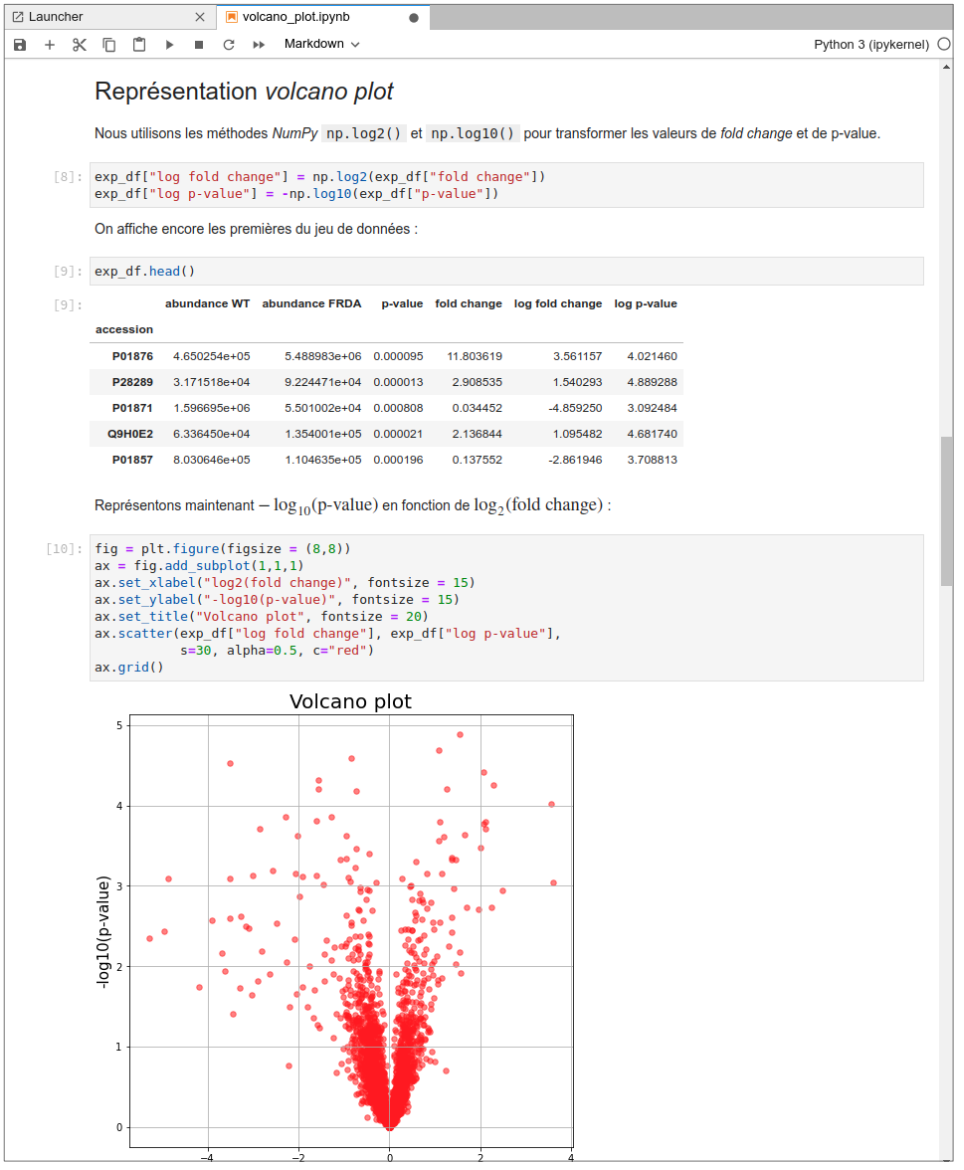
Retours utilisateurs (enseignants)

« PLASMA m'a permis de mettre en place un TP d'analyses multidimensionnées ne dépendant ni de la puissance de machine des étudiants, **ni d'une installation de leur part**. L'interface jupyter Notebook permet une **scénarisation du TP** mêlant **réflexion scientifique** et utilisation d'un langage informatique sans trop de difficultés. »

Développements pédagogiques

Notebooks :

- Exploration narrative, interactive
  et itérative des données
- Pédagogie active
- Développement
  de l'autonomie des étudiants



Conclusion

Solution opérationnelle
+250 utilisateurs, 15 UE

Maintenir les enseignements en distanciel
(Covid-19)

Utilisée par d’autres établissements
Université Rouen Normandie, CNAM

Prochaines étapes

Correction automatique des *notebooks* (2022)

Amélioration de l’interface (2023)

Dissémination (*open source*) :
Singapour (2022/23), UVSQ-Paris Saclay (2022)...