



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**

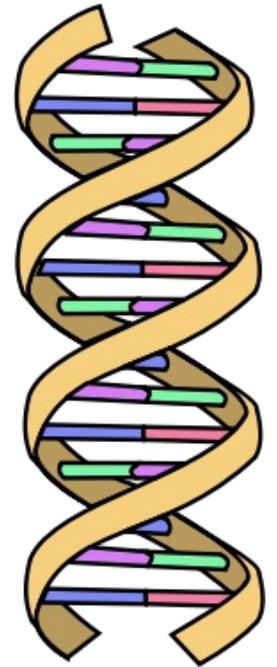
 Université
Paris Cité

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)



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 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



volcano_plot.ipynb 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()
```

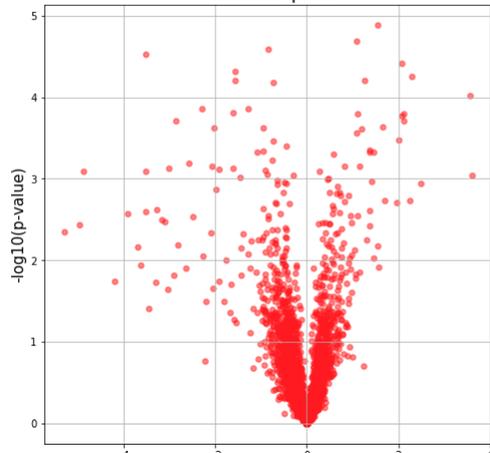
```
[9]:
```

	abundance WT	abundance FRDA	p-value	fold change	log fold change	log p-value
accession						
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()
```

Volcano plot



Cahier des charges (2018)

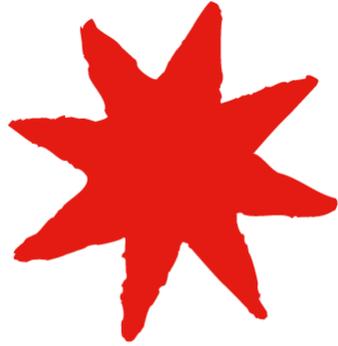
1. Puissance de calcul
(de 1 à 8 coeurs, de 2 à 32 Go RAM / étudiant)
2. Accès en dehors du réseau de l'université
3. Interface « intuitive »
(éviter le terminal, préférer une interface web et des *notebooks*)
4. Solution polyvalente pour différents types d'enseignements
5. Déploiement et administration « 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)



Région
île de France



200 k€

Serveurs + prestation développement + communication

Implémentation et développement (2020)

Configuration matérielle :

2 x Dell R840

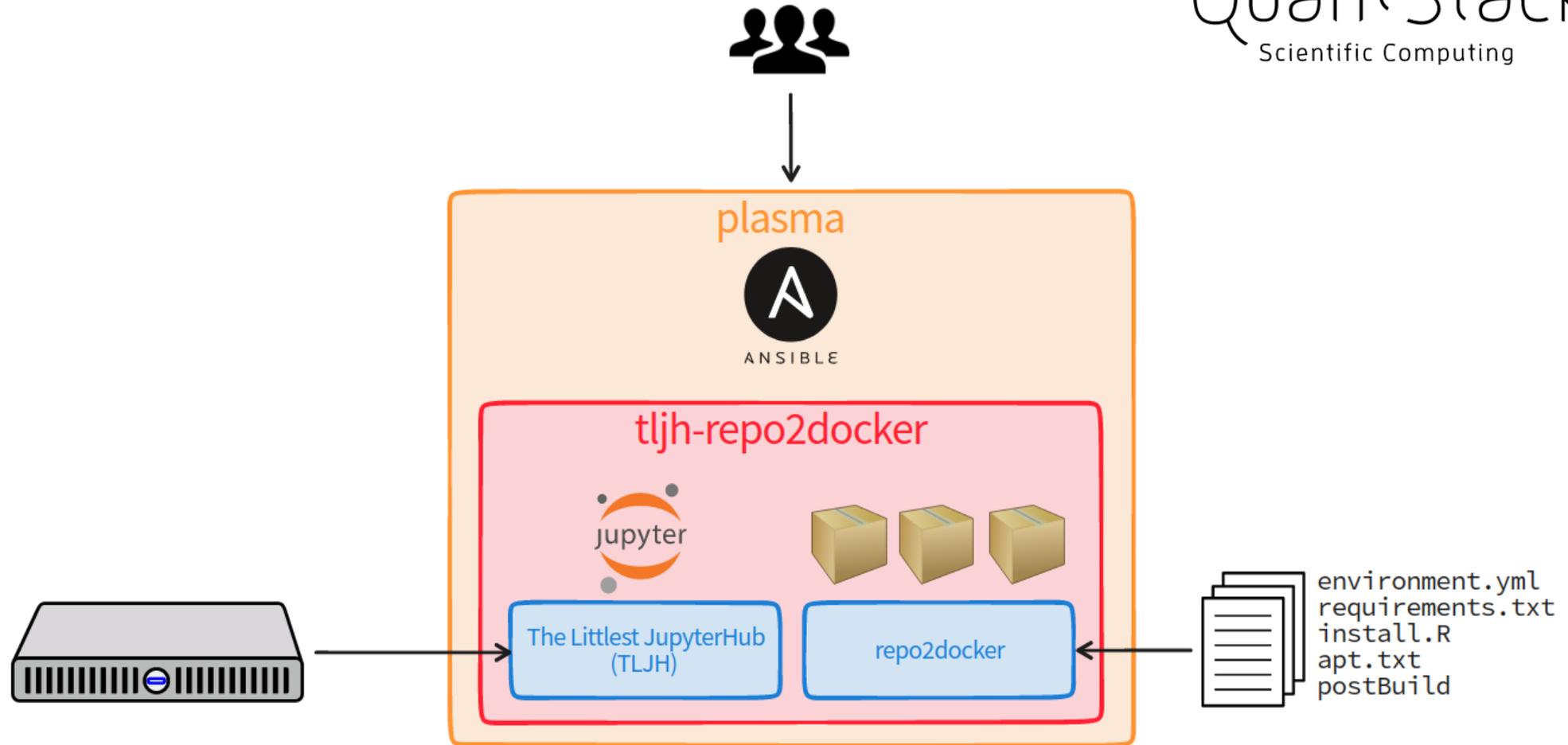
- 80 coeurs
- 768 Go RAM
- 30 To stockage

Prérequis techniques :

- Ubuntu 18.04 / 20.04
- Accès SSH et HTTPS

Implémentation et développement (2020)

QuantStack
Scientific Computing



Implémentation et développement (2020)



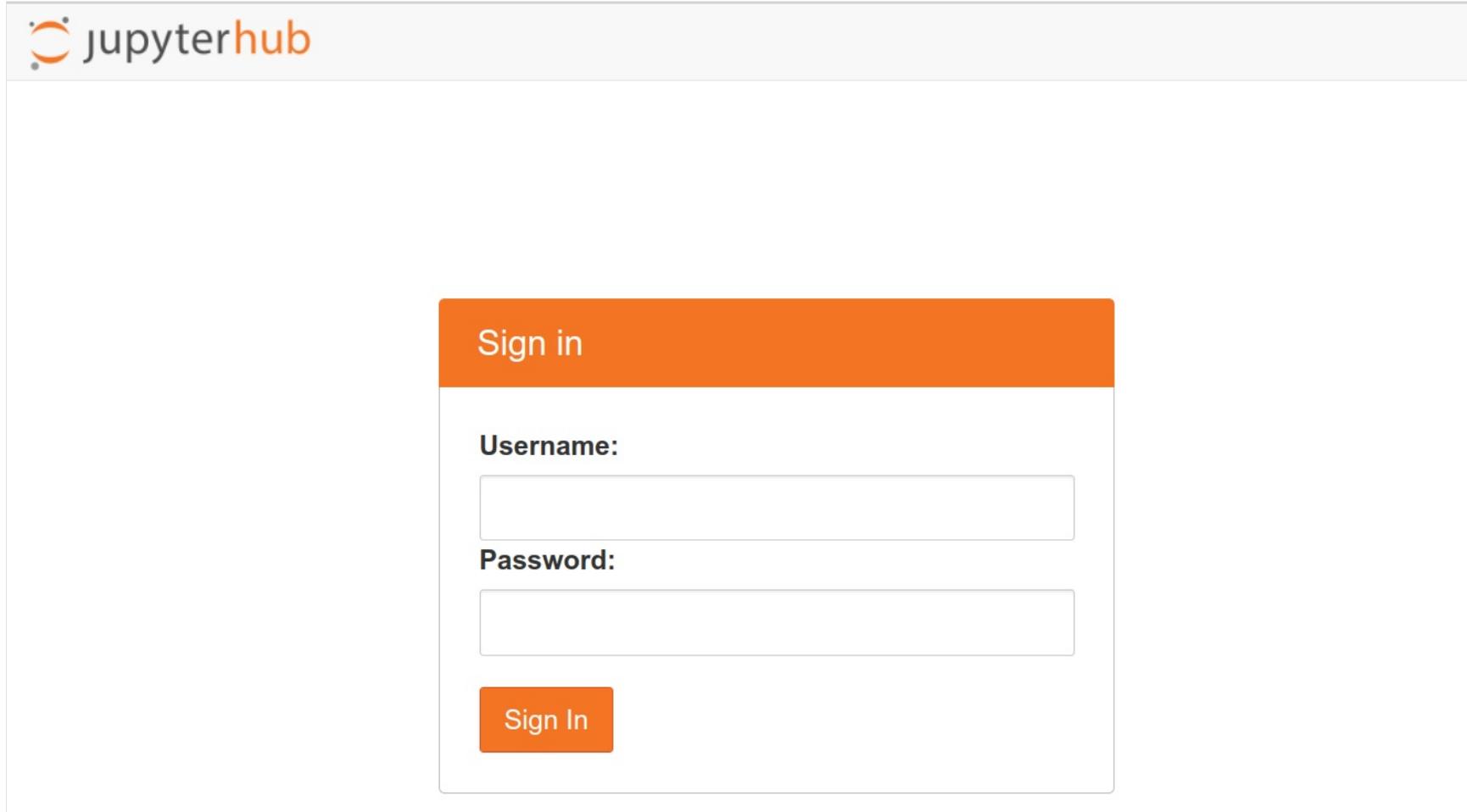
tljh-repo2docker

- 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



The image shows a screenshot of the JupyterHub login interface. At the top left, the JupyterHub logo is displayed, consisting of an orange circle with a white 'C' shape inside, followed by the text 'jupyterhub' in a sans-serif font. Below the logo, the main content area is white. In the center, there is a white rectangular box with a thin grey border. The top of this box is an orange bar with the text 'Sign in' in white. Below this bar, the text 'Username:' is followed by a white input field with a thin grey border. Below the input field, the text 'Password:' is followed by another white input field with a thin grey border. At the bottom of the box, there is an orange button with the text 'Sign In' in white.

jupyterhub

Sign in

Username:

Password:

Sign In

Choix des environnements

Server Options

- m2meg-rnaseq-practicals3to5-bash**
Repository: <https://github.com/Scaburet/M2MEG-RNaseq-TP3to5-bash.git>
Reference: [master](#)
Memory Limit (GB): **6**
CPU Limit: **10**
- plasma-demo**
Repository: <https://github.com/Scaburet/Plasma-demo.git>
Reference: [master](#)
Memory Limit (GB): **4**
CPU Limit: **10**
- m1meg-ue1-tp6-r**
Repository: <https://github.com/Scaburet/M1MEG-UE1-TP6-R.git>
Reference: [master](#)
Memory Limit (GB): **6**
CPU Limit: **4**
- m1meg-ue1-tp5-r**
Repository: <https://github.com/Scaburet/M1MEG-UE1-TP5-R.git>
Reference: [master](#)
Memory Limit (GB): **6**
CPU Limit: **4**

Construction d'un environnement

The image shows the JupyterHub interface with a 'Create Environment' dialog box open. The background is a table of existing environments with columns for Name, Repository URL, CPU Limit, Status, and an Add/Remove button. The dialog box has the following fields:

- Repository URL:** An empty text input field.
- Reference (git commit):** A text input field containing 'HEAD'.
- Name of the environment:** An empty text input field. Below it, there is an example: 'Example: course-python-101-B37' and a note: 'If empty, a name will be generated from the repo URL'.
- Memory Limit (GB):** A text input field containing '2'.
- CPU Limit:** A text input field containing '2'.

At the bottom of the dialog box, there are two buttons: 'Cancel' and 'Create Environment'.

Name	Repository URL	CPU Limit	Status	Action
pir_g2_gwas	https://git...		✓	Remove
pir_g2_gwas_test1	https://git...		✓	Remove
pass_rs_ue3_bistats	https://git...		✓	Remove
min_ue3_test_8	https://git...		✓	Remove
min_ue3_test_5	https://git...		✓	Remove
min_ue3_ai_test	https://git...		✓	Remove
l3meg-gh-tp2022	https://git...		✓	Remove
l3meg-gh-tp-2022	https://git...		✓	Remove
m1meg-score-polygen	https://git...		✓	Remove
meg_m1_gb_r	https://git...		✓	Remove
geno_bioinfo_r_test7	https://git...		✓	Remove
meg-m1-bt-rstudio	https://git...	main	✓	Remove

Modèles d'environnements

The screenshot shows the GitHub interface for the repository 'plasmabio/template-python'. At the top, there are navigation tabs for Code, Issues (1), Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation, there are buttons for 'Go to file', 'Add file', 'Code', and a prominent green 'Use this template' button. The main content area displays a commit history table and a file browser for 'README.md'. The commit history table lists the following entries:

File	Commit Message	Time Ago
binder	Stick to JupyterHub 1.x for now	6 days ago
.gitignore	Initial commit	2 years ago
LICENSE	Initial commit	2 years ago
README.md	Update Binder link	4 months ago
example-notebook.ipynb	Update repo url in notebook	22 days ago

The file browser shows the 'README.md' file with the heading 'Template materials in Python'. On the right side, the 'About' section provides details: 'Plasma template environment for Python', tags for 'python' and 'jupyter', 'Readme', 'BSD-3-Clause License', '1 star', '3 watching', and '7 forks'. The 'Contributors' section lists 'jtpio Jeremy Tuloup' and 'pierrepo Pierre Poulain'.

<https://github.com/plasmabio/template-python>

<https://github.com/plasmabio/template-r>

<https://github.com/plasmabio/template-bash>

Définition des permissions

pir_g2_gwas

Add Group

teachers Remove pir Remove

pir_g2_gwas_test1

Add Group

teachers Remove

pass_rs_ue3_bistats

Add Group

pass Remove teachers Remove

min_ue3_test_8

Add Group

teachers Remove

Interface JupyterLab

The image displays the JupyterLab interface. At the top, a menu bar includes File, Edit, View, Run, Kernel, Tabs, Settings, and Help. To the right of the menu, system information is shown: 'pass_rs_ue3_bistats', CPU usage at 0%, and memory usage at 108 / 6144 MB.

The left sidebar contains a file browser with a search bar labeled 'Filter files by name'. Below the search bar, the file list shows the following items:

- binder
- data
- bioinfo_1.ipynb
- LICENSE
- README.md

The main area is titled 'Launcher' and is divided into three sections:

- Notebook:** This section contains five icons for launching different environments: Python 3 (ipykernel), Bash, R, RStudio [↗], and Shiny [↗].
- Console:** This section contains three icons for launching different environments: Python 3 (ipykernel), Bash, and R.
- Other:** This section contains five icons for launching different file types: Terminal, Text File, Markdown File, Python File, and R File.

On the right side of the Launcher panel, there are two gear icons for settings.

Interfaces d'analyse JupyterLab / RStudio

The JupyterLab interface shows a file browser on the left with a search bar and a list of files including 'LICENSE' and 'README.md'. The main area displays a notebook cell with the following text:

-> Votre réponse...

2.2.B. Lister le contenu de n'importe quel répertoire

Pour lister le contenu d'un répertoire autre que le répertoire de travail, il suffit d'utiliser la commande `ls` (suivie de ses éventuels arguments) suivie du **chemin absolu** ou **relatif** de ce répertoire.

- le **chemin absolu** commence par un `/` et toute l'adresse
- le **chemin relatif** est positionné par rapport à votre répertoire de travail

- Listez le contenu du répertoire `/srv/data/pass-r`

```
[ ]: # cell 8  
# votre code
```

- Question 2 Quels droits avez-vous sur ces fichiers/répertoires ?

-> Votre réponse...

Ce répertoire contient des données que nous utiliserons :

The RStudio interface shows an R script editor with the following code:

```
27 #  
28 library(Biostrings)  
29  
30 # Create random DNA sequence of 10 nucléotides  
31 seq = sample(DNA_ALPHABET[1:4], size=10, replace=TRUE)  
32 seq = paste(seq, collapse="")  
33 seq  
34  
35  
36 #-----  
37 # Diagram  
38  
39 # https://cran.r-project.org/web/packages/diagram/index.html  
40 #-----  
41 library(diagram)  
42  
43 example(plotweb)  
44  
45  
46 #-----  
47 # Session info  
48 # sessionInfo()  
49 #>  
50:1 (Untitled) R Script
```

The console shows the R version and platform information:

```
R 4.1.2 .-./  
Copyright (C) 2021 The R Foundation for Statistical Computing  
Platform: x86_64-conda-linux-gnu (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.
```

The Environment pane on the right shows 'Global Environment' and 'Environment is empty'. The Files pane at the bottom shows a directory tree with folders like 'bus-dashboard', 'cours-bash-test', 'cours-r-intro', 'cours-unix', 'demo_r', 'introduction_plasma', 'm1meg_ue5_tp1-unix-1', 'm1meg-score-polygen', and 'm2meg-cran-practical3to5-bash'.

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. ».

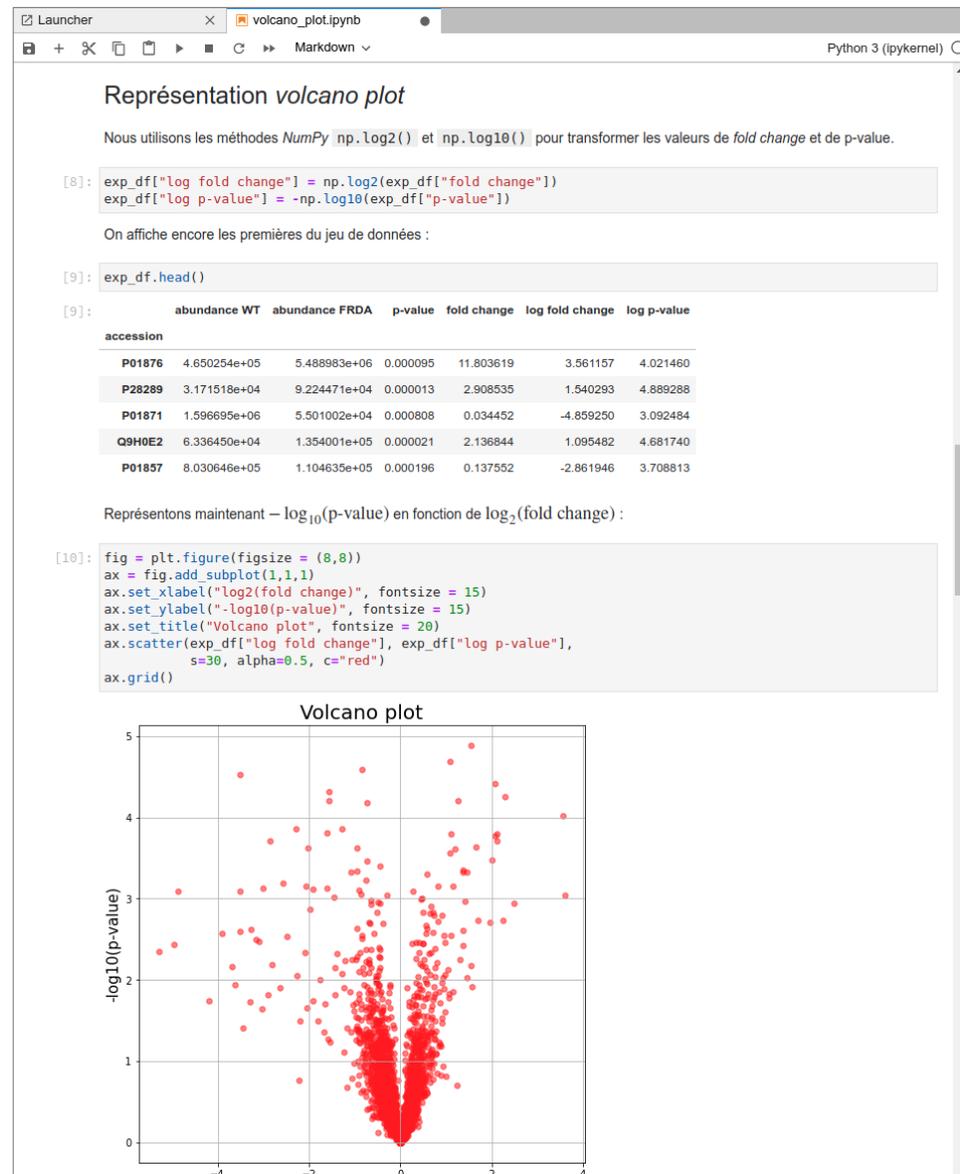
Retours utilisateurs (enseignants)

« 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)...