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

Jérémy Tuloup, Claire Vandiedonck, Sandrine Caburet, Pierre Poulain

▶ **To cite this version:**

Jérémy Tuloup, Claire Vandiedonck, Sandrine Caburet, Pierre Poulain. Plasma : plateforme d'e-learning pour l'analyse interactive de données. Journées Réseaux de l'Enseignement et de la Recherche (JRES), May 2022, Marseille, France. hal-03563658v1

HAL Id: hal-03563658

https://hal.science/hal-03563658v1

Submitted on 9 Feb 2022 (v1), last revised 10 May 2022 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Jérémy Tuloup
QuantStack
55 rue de Paris
94340 Joinville-le-Pont

Claire Vandiedonck
Institut Necker Enfants Malades & Université de Paris
156 rue de Vaugirard
75015 Paris

Sandrine Caburet
Institut Jacques Monod & Université de Paris
15 rue Hélène Brion
75013 Paris

Pierre Poulain
Institut Jacques Monod & Université de Paris
15 rue Hélène Brion
75013 Paris

Résumé

L’objectif de Plasma (https://plasmabio.org/) est de former des étudiants à l'exploration et à l'analyse de données massives, principalement issues d’expériences de génomique en biologie.

Ce projet repose sur l'utilisation de notebooks Jupyter, véritables carnets électroniques interactifs, qui intègrent du code informatique, du texte, des équations mathématiques et la visualisation des résultats d'analyse sous forme de graphiques ou de tableaux. Cette technologie est en train de devenir un standard pour l'analyse de données.

Ces notebooks sont hébergés sur un serveur informatique performant basé sur l’écosytème open source Jupyter. Nous avons développé une évolution de cette technologie qui permet aux enseignants de créer des environnements informatiques adaptés à chaque enseignement.

Les étudiants peuvent se connecter à distance quand et d’où ils le souhaitent pour effectuer leurs analyses dans un environnement convivial et performant.

Nous avons utilisé une première fois Plasma de septembre 2020 à avril 2021, pour des cours de programmation Python, de programmation R et d’analyses de données génomiques. Malgré des conditions sanitaires difficiles, nos étudiants ont pu poursuivre leurs enseignements pratiques grâce à cette solution, qu’ils ont appréciée et plébiscitée. Nous utilisons depuis lors Plasma pour de nombreux enseignements.

Ce projet est réalisé en collaboration avec QuantStack, une société fortement impliquée dans le développement de l'écosystème Jupyter. Plasma est open source, documenté et librement accessible.

Mots-clefs

Jupyter, notebook, analyse de données, enseignement, pédagogie active, génomique, open source, Plasma

1 Introduction : besoin initial

Claire Vandiedonck, Sandrine Caburet et Pierre Poulain sont trois enseignants-chercheurs à Université de Paris. Ils interviennent dans des enseignements de génétique, génomique et bioinformatique.

En 2016, dans le cadre d'enseignements du magistère européen de génétique d'Université de Paris, ils ont développé un enseignement de bioinformatique et génomique en première année de master. Le programme de cet enseignement comportait une introduction à Linux et à la ligne de commande, à la programmation Bash et au langage de traitement statistique R ainsi qu'à l'analyse de données haut-débit en génomique (RNA-seq). Cet enseignement, à l'époque optionnel, concernait une vingtaine d'étudiants biologistes sans compétences préalables en bioinformatique. Dans un premier temps, cet enseignement utilisait un serveur de récupération doté de 8 cœurs et 8 Go de mémoire vive.

Au fur et à mesure des années, nos besoins ont évolué :

1. Nous avions tout d'abord besoin de beaucoup plus de puissance. Les étudiants parvenaient difficilement à analyser un jeu de données, même réduit. Nous souhaitions désormais analyser un jeu de données issu de conditions réelles, nécessitant plusieurs cœurs et giga-octets de mémoire vive pour être traité. Parallèlement à cela, cet enseignement est devenu obligatoire et le nombre d'étudiants concernés est passé de 18 à 50.
2. L'accès au serveur était restreint au réseau de l'université et les étudiants ne pouvaient pas travailler sur leur projet depuis chez eux. Nous avions besoin que les étudiants puissent accéder plus facilement à cette ressource, et ce, depuis n'importe quel ordinateur.
3. Pour nos étudiants sans formation initiale en bioinformatique, le terminal et la ligne de commandes ne constituaient pas une interface très attrayante. Nous avions besoin d'interfaces de connexion et d'utilisation plus engageantes et accessibles aux étudiants.
4. Au-delà d'une interface conviviale et simple d'accès. Nous souhaitions que cette même interface puisse être utilisée pour d'autres enseignements, toujours de bioinformatique ou d'analyse de données, mais pouvant nécessiter d'autres langages de programmation, ainsi que des bibliothèques ou des logiciels différents.

2 Solution technique : l'écosystème Jupyter

Les deux premiers besoins pouvaient être satisfaits relativement simplement avec un serveur physique ou virtuel suffisamment puissant, accessible depuis l'extérieur. Les solutions cloud « élastiques » comme Kubernetes ont rapidement été éliminées, car posant un double problème de facturation et de compétences techniques pour les administrer.

Pour le troisième besoin d'une interface conviviale, nous nous sommes rapidement orientés vers le projet Jupyter[1] et son écosystème [1]. Ce projet bénéficie d'une communauté très active et propose un écosystème riche composé d'outils et de logiciels open source, parmi lesquels :

– JupyterHub : un portail de connexion web qui supporte plusieurs modes d'authentification des utilisateurs (PAM, SAML..)

– Jupyter Server : un serveur web qui propose l'accès à l'interface d'analyse de données JupyterLab, mais également à l'environnement de développement RStudio, spécifique au langage R.

– JupyterLab : une interface de traitement et d'analyse de données très complète. Cette interface propose l'édition de *notebooks* Jupyter dans une cinquantaine de langages différents (dont Python, Bash, R, C++), un terminal Unix, l'édition de fichiers textes, la visualisation d'images et de fichiers tabulés au format CSV et TSV. Enfin, un explorateur de fichiers permet d'organiser fichiers et répertoires simplement.

L'ensemble de ces éléments constitue un écosystème homogène, cohérent et accessible par un simple navigateur web.

Au-delà de ces interfaces de connexion et de traitement de données, particulièrement conviviales, le cœur de l'écosystème Jupyter repose sur les *notebooks* Jupyter. Ces *notebooks* sont de véritables carnets électroniques interactifs ou « bloc-code » [2], qui intègrent du code informatique, du texte, des équations mathématiques et la visualisation des résultats d'analyse sous forme de graphiques ou de tableaux.

En quelques années, cette technologie est progressivement devenue un standard pour l'analyse de données [1,3,4]. D'autres types de *notebooks*, comme les documents R Markdown, sont également pertinents pour l'analyse de données, notamment avec le langage R dans l'interface RStudio. En proposant de « narrer » une analyse de données, les *notebooks* implémentent la notion de « *literate programming* » telle qu'elle a été établie par Knuth [5]. Cet aspect déclaratif et explicite du processus d'analyse [6] confère aux *notebooks* un rôle important dans la reproductibilité des analyses [1,7,8]. L'exemple de Wang et Ma'ayan avec l'analyse génomique du virus Zika en 2016 [9] et le *notebook* correspondant[2] constitue un prototype très intéressant sur ce point.

Quant au quatrième et dernier besoin portant sur une interface polyvalente et pouvant être utilisée pour plusieurs enseignements différents, l'écosystème Jupyter et particulièrement le projet « *The Littlest JupyterHub* »[3] ne pouvaient y répondre en l'état et nécessitaient un développement spécifique.

3 Le projet Plasma

Afin de répondre aux besoins mentionnés précédemment, nous avons créé le projet Plasma[4] pour « Plateforme d'e-Learning pour l'Analyse de données Scientifiques MAssives ». L'objectif dans un

1 https://jupyter.org/

2 https://nbviewer.jupyter.org/github/maayanlab/Zika-RNAseq-Pipeline/blob/master/Zika.ipynb

3 https://tljh.jupyter.org/

4 https://plasmabio.org/

premier temps était de développer un prototype pour l'enseignement de l'analyse de données en biologie, et plus particulièrement en génomique.

3.1 Financement du projet

Pour financer ce projet, nous avons répondu à l'appel d'offre des « Trophées franciliens de l'innovation numérique dans le supérieur » organisé par la région Île-de-France en 2018 et dont nous avons été lauréats. Nous avons ainsi obtenu 75 k€.

En 2019, nous avons également obtenu 62 k€ de l'IDEx de l'Université de Paris, ainsi que 10 k€ de l'École Universitaire de Recherche Génétique et Épigénétique Nouvelle École (EUR GENE[5]) et 9,3 k€ du diplôme universitaire « Création, analyse et valorisation de données biologiques omiques » (DU omiques[6]) d'Université de Paris.

3.2 Implémentation et développements

Nous avons acheté deux serveurs Dell via le marché public de l'enseignement supérieur et la recherche Matinfo4. Les machines sont des serveurs Dell R840, disposant chacun de 80 cœurs Intel Xeon 6230 hyperthreadés, 768 Go de RAM et de 30 To de stockage. Commandés fin 2019, les serveurs ont été livrés début 2020.

Nous avons demandé à la société QuantStack de développer une adaptation d'une implémentation déjà existante d'un JupyterHub sur un serveur physique ou virtuel, appelée « *The Littlest JupyterHub* », afin de pouvoir construire et utiliser des environnements de travail étanches. La société QuantStack, créée et dirigée par Sylvain Corlay, est spécialisée dans le développement open source de solutions autour du projet Jupyter (système de *dashboard* Voilà, noyau C++ Xeus…). Elle est aussi fortement impliquée dans l'écosystème scientifique : mamba (alternative à conda), conda-forge…

D'emblée nous avons souhaité que tous les développements logiciels liés au projet Plasma soient diffusés sous licence *open source*, de façon à pouvoir contribuer à l'écosystème Jupyter existant.

Jérémy Tuloup, développeur scientifique chez QuantStack a travaillé sur le projet de mars à juin 2020, en plein confinement. Il a produit deux développements originaux disponibles en *open source* sur GitHub :

– tljh-repo2docker (https://github.com/plasmabio/tljh-repo2docker) : une extension de The Littlest JupyterHub qui propose aux utilisateurs des environnements virtuels autonomes constitués de conteneurs Docker. Les images Docker elles-mêmes sont construites à partir de l'outil repo2docker[7] qui est notamment utilisé par le projet Binder[8].

– plasma (https://github.com/plasmabio/plasma) : un ensemble de scripts d'automatisation, sous la forme de *playbooks* Ansible, utilisés pour déployer Plasma et tljh-repo2docker sur un serveur physique ou virtuel.

5 https://eur-gene.u-paris.fr/

6 https://omics-school.net/

7 https://repo2docker.readthedocs.io/en/latest/

8 https://elifesciences.org/labs/8653a61d/introducing-binder-2-0-share-your-interactive-research-environment

Nous avons déployé Plasma sur les deux serveurs physiques en juin 2020. Nous avons installé le système d'exploitation Ubuntu 20.04 LTS sur un volume en RAID 1 constitué de deux disques SSD de 1 To. Les répertoires utilisateurs ainsi que les données sont stockés sur un volume en RAID 5 constitué de disques SATA représentant au total 29 To utiles.

L'outil Plasma lui-même est déployé avec plusieurs *playbooks* Ansible (disponibles sur https://github.com/plasmabio/plasma) qui mettent à jour le système, installent Docker et The Littlest JupyterHub avec son extension *tljh-repo2docker*, *repo2docker*. Enfin ils installent et configurent un *proxy web* et la couche HTTPS via Let's Entcrypt.

La création des utilisateurs, la définition de leurs droits et de leur quota d'espace disque passe également par des *playbooks* Ansible. L'authentification des utilisateurs est basée sur l'authentification PAM Unix.

Chaque enseignant peut définir son propre environnement logiciel et y associer les ressources informatiques nécessaires. Notre expérience montre qu'un CPU et 1 Go de RAM suffisent pour des enseignements d'introduction à la programmation ou à Unix. Pour certains traitements ou analyses plus conséquents, des environnements avec 8 CPU et 8 ou 16 Go de RAM sont parfois nécessaires (par exemple pour de l'analyse RNA-seq en master). Il n'existe cependant pas de règle simple pour déterminer le dimensionnement optimal d'un environnement. La charge globale du serveur résulte du nombre d'environnements chargés simultanément et de la taille de ces environnements. Jusqu'à présent nous n'avons pas été confronté à des serveurs sur-chargés.

Les tous premiers enseignements ont débuté en septembre 2020. Dans le contexte de pandémie de la Covid-19 et avec les confinements successifs, Plasma nous a permis d'assurer la continuité pédagogique de l'intégralité de nos enseignements pendant l'année universitaire 2020-2021 et durant l'année en cours.

3.3 Développements scientifiques

Le projet Plasma comprend également un volet de développements scientifiques afin d'adapter deux outils couramment utilisés en bioinformatique à leur utilisation dans un *notebook* Jupyter.

– ipyigv (https://github.com/QuantStack/ipyigv) : adaptation de l'outil de visualisation d'alignement génomique IGV. Une première version du projet a été livrée début 2021 par Jean-David Harrouet[9].

– ipycytoscape (https://github.com/cytoscape/ipycytoscape) : adaptation de l'outil de visualisation et de manipulation de réseau Cytoscape. Dès avril 2020, une première version développée par Mariana Meireles était disponible[10]. Devant le succès du projet dans la communauté, QuantStack a, avec notre accord, transféré la gestion du projet au consortium Cytoscape en mai 2021[11].

9 https://blog.jupyter.org/genomic-data-representation-in-jupyter-c57a5bb518d6

10 https://blog.jupyter.org/interactive-graph-visualization-in-jupyter-with-ipycytoscape-a8828a54ab63

11 https://mari-meir.medium.com/jupyter-%EF%B8%8F-%EF%B8%8F-cytoscape-e2e77be8e0f9

3.4 Retour des utilisateurs et bilan de l'utilisation

À l'issue du premier semestre de l'année universitaire 2020-2021, nous avons interrogé les étudiants et les enseignants sur leurs utilisations de Plasma.

Les utilisateurs ont attribué le score de 4,4 à la proposition « Le serveur Plasma est simple d'utilisation » et le score de 4,7 à la proposition « Le serveur Plasma m'a permis de travailler à distance », où un score de 1 signifie « pas du tout » et un score de 5 « complètement ».

Des enseignants nous ont également laissé quelques témoignages :

– « *J'ai utilisé le serveur PLASMA dans le cadre de séances d'initiation à Unix. La création de l'environnement a été très simple et a complètement contourné l'épineux problème de l'installation d'un système Unix fonctionnel sur les ordinateurs personnels des étudiants. La prise en main de l'environnement PLASMA s'est faite rapidement et de manière assez intuitive par l'enseignant et les étudiants. La possibilité d'utiliser la console Unix ou les notebooks permet de varier les angles pédagogiques. Nombre de mes enseignements pourraient être à terme basculés sur PLASMA si l'infrastructure le permet.* » (B.T.)

– « *L'utilisation de PLASMA m'a permis de mettre en place un TP d'analyses multi dimensionnées 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.* » (F.F.)

Tout comme certains étudiants :

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

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

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

Entre septembre 2020 et juin 2021, plus de 110 étudiants et 17 enseignants ont utilisé Plasma dans une dizaine d'enseignements depuis la 3e année de licence jusqu'à la seconde année de master.

3.5 Développements pédagogiques

Outre l'aspect purement technique et « pratique » de la plateforme, nous sommes également très satisfaits des aspects pédagogiques apportés par Plasma. L'utilisation et la construction de *notebooks* rendent les étudiants acteurs de leurs apprentissages. Ils explorent et analysent leurs données de façon interactive et itérative. Ce type d'approche est parfaitement en ligne avec l'idée de pédagogie active [10,11], qui favorise une acquisition durable des savoirs et des compétences chez les étudiants.

Concrètement, nous avons fait évoluer nos enseignements pour qu'ils intègrent des *notebooks* du premier au dernier cours. Les *notebooks* sont devenus des supports d'information et remplacent la majorité de nos diaporamas ou énoncés de travaux pratiques. Ils sont aussi devenus des documents d'exercices avec la possibilité de guider les étudiants tout le long d'une analyse de données, en fournissant notamment des exemples, des explications plus détaillées, ou bien encore une

progression dans l'analyse de données. En proposant aux étudiants de réaliser, par exemple, leur projet au format *notebook* Jupyter dans JupyterLab ou au format Rmarkdown dans RStudio, nous leur laissons une liberté qui contribue à leur motivation.

Enfin, les *notebooks* et l'infrastructure de Plasma dans son ensemble nous ont permis de passer nos enseignements en distanciel très facilement. La crise sanitaire du Covid-19 nous a malheureusement donné raison. Au-delà d'un enseignement à distance subi, Plasma nous offre la possibilité d'un véritable enseignement hybride avec des enseignements qui peuvent être, par exemple, initiés dans les locaux de l'université et être naturellement poursuivis ou répétés avec d'autres paramètres à l'extérieur (à la maison ou n'importe où ailleurs).

4 Évolution et dissémination

La simplicité et la polyvalence de la plateforme Plasma fait qu'elle est d'ores et déjà utilisée au-delà de nos propres enseignements à l'UFR Sciences du vivant. Elle est, par exemple, utilisée à l'UFR de Médecine d'Université de Paris (Parcours d'Initiation Recherche : « Bioinformatique - Génomique », Mineure de la PASS : « Recherche en Santé »). Nous sommes également en interaction avec nos collègues de l'Université de Rouen pour que Plasma y soit déployé pour être utilisé par les étudiants de l'UFR de Santé puis par ceux de l'UFR de Sciences.

Nous avons aussi remporté un appel à projet en décembre 2021 (46 k€) pour collaborer avec l'Université de Singapour et y déployer Plasma pour des enseignements de bioinformatique. Nous allons en particulier adapter Plasma pour la création massive de comptes étudiants et pour la correction automatique de *notebooks*.

Enfin, nous avons présenté Plasma à plusieurs congrès nationaux et internationaux : JupyterCon 2020 (création d'une vidéo de présentation de Plasma[12]), Journées Ouvertes en Biologie, Informatique et Mathématiques (JOBIM) 2021 et European Human Genetics Conference (ESHG) 2021. De façon plus informelle, nous avons aussi créé un site web pour le projet[13] et plusieurs articles de blog rédigés par l'équipe de QuantStack ont relaté les innovations techniques réalisées.

5 Conclusion

Le projet Plasma est né en 2018 d'un besoin d'accessibilité, de performance et de polyvalence. Malgré les nombreuses difficultés administratives et techniques que nous avons rencontrées pour initier, financer et déployer ce projet, Plasma est désormais opérationnel et remplit pleinement ses missions.

Le projet Plasma est entièrement *open source* et disponible sur la plateforme de développement GitHub[14]. Au delà de l'aspect purement technique, l'approche « *notebook* » proposée par Plasma possède des avantages pédagogiques qui favorisent l'engagement des étudiants.

12 https://www.youtube.com/watch?v=0KIMSPTMzVY

13 https://plasmabio.org/

14 https://github.com/plasmabio/plasma

Le pari de baser Plasma sur l'écosystème du projet open source Jupyter s'est révélé gagnant. Cet écosystème est aujourd'hui largement utilisé, depuis le projet Capytale[15] déployé dans l'éducation nationale ou Callisto[16] en sciences humaines et sociales, jusqu'aux nombreuses infrastructures de calcul haute performance en France (IFB[17]) et à l'étranger : DEEP Hybrid DataCloud[18] et SoBigData[19] (Europe), Syzygy[20] (Canada), NERSC[21] (USA), NUS[22] (Singapour).

15 https://capytale2.ac-paris.fr/

16 https://hnlab.huma-num.fr/blog/2021/05/26/callisto-un-demonstrateur-jupyter/

17 https://jupyterhub.cluster.france-bioinformatique.fr/

18 https://deep-hybrid-datacloud.eu/

19 http://www.sobigdata.eu/index

20 https://intro.syzygy.ca/

21 https://cs.lbl.gov/news-media/news/2021/project-jupyter-a-computer-code-that-transformed-science/

22 https://nusit.nus.edu.sg/services/hpc-newsletter/preview-jupyterhub-hpc/

Bibliographie

1. Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, et al. Jupyter Notebooks – a publishing format for reproducible computational workflows. Positioning and Power in Academic Publishing: Players, Agents and Agendas [Internet]. 2016. p. 87-90. Disponible sur: https://ebooks.iospress.nl/publication/42900

2. Perret A. Du notebook au bloc-code [Internet]. 2021 [cité 7 janv 2022]. Disponible sur: https://www.arthurperret.fr/blog/2021-06-11-du-notebook-au-bloc-code.html

3. Kery MB, Radensky M, Arya M, John BE, Myers BA. The Story in the Notebook: Exploratory Data Science using a Literate Programming Tool. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18 [Internet]. Montreal QC, Canada: ACM Press; 2018 [cité 5 juill 2020]. p. 1-11. Disponible sur: http://dl.acm.org/citation.cfm?doid=3173574.3173748

4. Somers J. The Scientific Paper Is Obsolete. The Atlantic [Internet]. 5 avr 2018; Disponible sur: https://www.theatlantic.com/science/archive/2018/04/the-scientific-paper-is-obsolete/556676/

5. Knuth DE. Literate Programming. The Computer Journal. 1984;27:97-111.

6. Granger BE, Perez F. Jupyter: Thinking and Storytelling With Code and Data. Comput Sci Eng. 2021;23:7-14.

7. Rule A, Birmingham A, Zuniga C, Altintas I, Huang S-C, Knight R, et al. Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. Lewitter F, éditeur. PLOS Computational Biology. 2019;15:e1007007.

8. Beg M, Taka J, Kluyver T, Konovalov A, Ragan-Kelley M, Thiery NM, et al. Using Jupyter for Reproducible Scientific Workflows. Comput Sci Eng. 2021;23:36-46.

9. Wang Z, Ma'ayan A. An open RNA-Seq data analysis pipeline tutorial with an example of reprocessing data from a recent Zika virus study. F1000Research. 2016;5:1574.

10. Freeman S, Eddy SL, McDonough M, Smith MK, Okoroafor N, Jordt H, et al. Active learning increases student performance in science, engineering, and mathematics. Proceedings of the National Academy of Sciences. 2014;111:8410-5.

11. Davies A, Hooley F, Causey-Freeman P, Eleftheriou I, Moulton G. Using interactive digital notebooks for bioscience and informatics education. Ouellette F, éditeur. PLoS Comput Biol. 2020;16:e1008326.