



Introduction à la journée

Damien Magoni¹

La Société informatique de France (SIF) a organisé le 10 mai 2021 une journée de présentations et d'échanges sur la reproductibilité de la recherche², un thème qui a pris de plus en plus d'ampleur depuis une dizaine d'années. La journée a réuni huit orateurs sur des thèmes centrés autour de cette problématique. Cette journée s'est tenue en visio-conférence et les interventions ont été enregistrées³.

Après le mot de bienvenue d'Yves Bertrand, président de la SIF, un premier exposé introductif intitulé « La recherche reproductible : c'est quoi ? pourquoi en faire ? comment ? » a été donné par Christophe Pouzat (université de Strasbourg/CNRS). Sa présentation a commencé par un court historique des pratiques que désigne la « recherche reproductible ». Christophe a ensuite présenté certains des outils qui lui ont permis de la pratiquer concrètement depuis 2005. Il a enfin exposé les gains considérables que cette pratique permet de réaliser dans le cadre de projets impliquant des chercheurs de plusieurs disciplines. Chercheur CNRS à l'IRMA de Strasbourg, Christophe Pouzat est neurophysiologiste et travaille sur l'analyse de données.

L'exposé suivant intitulé « Archiver et référencer le code source des logiciels de recherche pour la science ouverte et la reproductibilité » a été donné par Roberto Di Cosmo (université Paris Diderot/Inria) et a concerné l'archive Software Heritage. Le logiciel est en effet un pilier fondamental de la recherche scientifique moderne, et il est primordial d'archiver et de référencer correctement les codes sources du logiciel utilisé dans toutes les activités de recherche. Dans sa présentation, Roberto Di Cosmo

1. Professeur en informatique, université de Bordeaux.

2. Les supports des présentations sont téléchargeables depuis la page web de l'évènement <https://www.societe-informatique-de-france.fr/journee-reproductibilite>.

3. Les vidéos sont accessibles via <https://vimeo.com/showcase/8652937>.

a montré comment utiliser l'infrastructure Software Heritage pour archiver de façon transparente tous les codes sources des logiciels et comment obtenir et utiliser les identifiants intrinsèques, spécifiquement conçus pour référencer le code source d'une manière qui facilite la reproductibilité à long terme. Roberto Di Cosmo est professeur d'informatique à l'université Paris Diderot, membre du comité d'orientation pour la science ouverte en France, et actuellement détaché chez Inria. Suivant de près l'impact de l'informatique sur la société, il prône depuis longtemps l'adoption du logiciel libre. En 2015, il a été à l'origine de Software Heritage, une initiative qui vise à construire l'archive universelle de tout le code source publiquement disponible, en partenariat avec l'UNESCO.

Konrad Hinsén du centre de biophysique moléculaire d'Orléans, nous a ensuite présenté ses travaux sur « La reproductibilité des calculs coûteux » qui indique comment assurer ou vérifier la reproductibilité d'un calcul qui occupe un centre de calcul pendant trois semaines ? Ou encore, comment faire si des contraintes techniques nous obligent à exécuter chaque étape d'un calcul sur un autre ordinateur ? Les outils standards pour la recherche reproductible n'ayant pas encore de réponses à ces problématiques, Konrad a indiqué comment le projet ActivePapers a tenté de relever ces défis, et avec quel degré de succès. Konrad Hinsén, travaille sur la structure et la dynamique des protéines avec les méthodes de la simulation moléculaire. Auteur de plusieurs bibliothèques scientifiques en Python, membre fondateur du projet NumPy et enseignant pour Software Carpentry, Konrad est co-auteur d'un MOOC sur la recherche reproductible avec Arnaud Legrand.

Le dernier exposé de la matinée a été consacré à « la randomisation, une solution aux difficultés de reproductibilité de mesures de performances de processeurs modernes ? » et a été présenté par Arnaud Legrand (université de Grenoble-Alpes/CNRS). Lorsque l'on parle de problèmes de reproductibilité de la recherche, on met souvent en avant des problèmes de provenance, de manque de contrôle du logiciel et de son environnement, de problèmes de stabilité numérique, etc. liés à un mauvais usage des statistiques. Les aspects mesures et expériences sont souvent assez peu évoqués, peut-être parce que les difficultés expérimentales sont un problème ancien et déjà assez bien cerné pour un certain nombre de disciplines (physique, biologie), ce qui n'est pas forcément le cas en informatique qui est une science relativement jeune. Les processeurs modernes sont des objets complexes avec des hiérarchies de caches au comportement parfois opaque, aux stratégies de vectorisation et de réordonnancement dynamique d'instruction obscures, dont la fréquence de fonctionnement varie au cours du temps pour s'adapter à la charge de travail et à la température du processeur, etc. Effectuer des mesures fiables sur ces processeurs peut vite devenir assez difficile, et obtenir des comportements « similaires » entre deux machines supposées identiques, voire sur la même machine à deux périodes différentes, peut être un véritable casse-tête. L'exposé a présenté des plans d'expériences robustes à adopter permettant de se prémunir contre ces problèmes ou de les

identifier. Arnaud a aussi présenté quelques « histoires d'horreur » et les limitations potentielles de la randomisation dans ce contexte. Chercheur au CNRS depuis 2004, Arnaud Legrand s'intéresse à l'étude des infrastructures informatiques distribuées, à grande échelle, utilisées pour le calcul scientifique. Cela recouvre les questions d'optimisation liées à l'exploitation de ces plateformes (ordonnancement, optimisation combinatoire, théorie des jeux) et les techniques d'évaluation de performances (simulation, visualisation, analyse statistique). Il est l'un des concepteurs du projet SimGrid, un outil de simulation open source dont le but est de faciliter la recherche dans le domaine de l'optimisation des systèmes parallèles et distribués. Depuis 5 ans, il participe à la promotion de meilleures pratiques expérimentales et à l'amélioration de la méthodologie scientifique à travers des tutoriels, des conférences, des cours, et un MOOC sur la recherche reproductible avec Konrad Hinsin.

L'après-midi a été composé de quatre exposés dont le premier intitulé « Reproductibilité computationnelle en sciences de la vie et workflows scientifiques : état des lieux et retour d'expérience » a été présenté par Sarah Cohen-Boulakia (université Paris-Saclay). Sa présentation a dressé le bilan des travaux du groupe de travail ReproVirtuFlow du GDR MaDICS qui s'intéresse à la reproductibilité des analyses de données bio-informatiques. De nombreux outils et familles de solutions existent aujourd'hui pour tendre vers une meilleure reproductibilité computationnelle des résultats bio-informatiques. Son équipe a testé ces outils dans le cadre de l'organisation de « reprohackathons », hackathons dont l'objectif est de reproduire un résultat publié. Son équipe a proposé notamment la définition de différents niveaux de reproductibilité d'une analyse et l'identification de caractéristiques clés des outils existants vis-à-vis de leur capacité à reproduire des résultats. Professeur à l'université Paris-Saclay dans l'équipe bio-informatique du laboratoire interdisciplinaire des sciences du numérique (LISN), Sarah Cohen-Boulakia est directrice du GDR MaDICS. Son domaine d'expertise porte sur la reproductibilité des analyses bio-informatiques et en particulier sur la provenance dans les workflows scientifiques.

L'exposé suivant, intitulé « Archiver, identifier, décrire et citer le code source : le dépôt de logiciel de recherche sur l'archive ouverte HAL », a été présenté par Morane Gruenpeter (Software Heritage). L'importance de déposer et partager nos logiciels de recherche nécessite de savoir comment le faire. Morane a présenté quatre types d'utilisation indispensables pour déchiffrer le dépôt logiciel et les bonnes pratiques associées à chacun. Le premier usage concerne l'archivage du code source dans HAL et sur Software Heritage, qui est une action nécessaire pour assurer l'accessibilité à long terme. Le deuxième usage concerne l'identification de la version spécifique avec un identifiant intrinsèque, une première étape pour conquérir le défi de la reproductibilité. Le troisième usage concerne la description du code source pour une meilleure compréhension du logiciel et pour permettre la découverte du logiciel sur des moteurs de recherche. Et enfin, le dernier concerne la citation du logiciel pour attribuer le crédit aux auteurs dans l'écosystème académique. Morane a

rejoint l'équipe Software Heritage en 2017. En 2018-2019, elle a continué son travail de recherche en collaboration avec le projet européen EU2020 CROSSMINER pour construire le Web sémantique des projets FOSS par la compilation des métadonnées existantes. Elle est membre active de plusieurs groupes de travail dans les domaines de la citation du logiciel, de la préservation numérique et du Web sémantique.

Nicolas Rougier (Inria Bordeaux) a ensuite présenté ReScience C à travers son exposé intitulé *Long term reproducibility*. ReScience C est un journal à comité de lecture en accès ouvert qui cible la recherche computationnelle et encourage la réplique de travaux de recherche déjà publiés. Ce journal a été co-fondé par Konrad Hinsén et lui-même. Il promeut des nouvelles implémentations open-source afin de s'assurer que la recherche initiale soit reproductible. Son comité éditorial a organisé le *Ten Years Reproducibility Challenge* qui consiste à inviter les chercheurs à essayer d'exécuter le code qu'ils avaient créé pour une publication scientifique faite il y a plus de 10 ans. Malgré sa facilité apparente, ce défi est plus complexe qu'il n'y paraît. Le premier problème pourrait d'ailleurs être celui de retrouver le code source à un moment où Software Heritage n'existait pas ! Chercheur en neuroscience computationnelle cognitive à Inria et à l'institut des maladies neurodégénératives de Bordeaux, Nicolas Rougier effectue des recherches portant sur les modèles computationnels du cerveau permettant la prise de décision, l'apprentissage et la cognition. Il s'intéresse aussi aux réseaux de neurones artificiels et à l'apprentissage automatique, ainsi qu'à la recherche reproductible, la visualisation scientifique et l'informatique graphique.

L'après-midi s'est clôturée sur un exposé intitulé « Environnements logiciels reproductibles et transparents avec GNU Guix » et présenté par Ludovic Courtès (Inria). La reproductibilité des expériences impliquant du logiciel est un enjeu scientifique majeur. Pourtant celle-ci se heurte souvent à la difficulté de répliquer mais aussi d'inspecter et de modifier de manière contrôlée les environnements logiciels de ces expériences. Une réponse populaire à ce problème est l'utilisation d'outils permettant de conserver les octets qui constituent l'environnement logiciel d'une expérience. À cette approche opaque, GNU Guix oppose une approche inspirée de la programmation fonctionnelle où un déploiement logiciel à partir de son code source est vu comme un « calcul » comme un autre, et où chacun de ces calculs est une fonction pure. Ludovic a présenté ces fondements et a montré en quoi Guix est une solution utilisable aujourd'hui pour définir des environnements logiciels reproductibles bit à bit. Il a de plus abordé les implications de cette approche sur les pratiques de publication scientifique. Ingénieur de recherche Inria, Ludovic Courtès a initié le développement de GNU Guix il y a huit ans. Depuis 2017, il s'intéresse à son utilisation dans le cadre du calcul intensif (HPC) et de la science reproductible avec le projet Guix-HPC. Ludovic est un adepte de la programmation fonctionnelle, notamment en Scheme avec GNU Guile auquel il a contribué.

Cette journée a eu un franc succès avec une moyenne tournant autour de 110 téléspectateurs environ et un pic à 156 téléspectateurs.