Un panorama de la recherche reproductible

Christophe Pouzat

Un panorama de la recherche reproductible

Christophe Pouzat
IRMA, Université de Strasbourg et CNRS
christophe.pouzat@math.unistra.fr

25 août 2021

1 Introduction

Qu'est-ce que la « recherche reproductible » et pourquoi en faire ?

La recherche reproductible a pour but de diminuer l'écart entre un idéal — les résultats de recherche publiés devraient être reproductibles — et la réalité — il est souvent difficile, même pour leurs auteurs, de reproduire des / leurs résultats. Concrètement, c'est une démarche qui consiste à fournir aux lecteurs d'articles, d'ouvrages, etc, l'ensemble des données et des programmes utilisés pour obtenir les résultats présentés *accompagnés d'une description algorithmique de la façon dont les programmes ont été appliqués aux données, ainsi que, si besoin, de l'environnement de calcul.*

Un élément implicite mais important de la définition précédente est que, dans la pratique, ce qui est entendu par « reproduction » est tout ce qui vient *après* la collecte des données — il serait donc plus judicieux de parler d'*analyse reproductible des données* —, mais comme l'approche requiert un *accès libre* à celles-ci, elles deviennent critiquables et comparables, *ce qui devrait améliorer la reproductibilité des données elles-mêmes.*

Du fait du caractère très explicite de la présentation des résultats que la recherche reproductible impose, considérée comme méthodologie, elle intéresse un public beaucoup plus large que celui des (enseignants-)chercheurs produisant des publications. Même si les résultats ne sont pas rendus publics, comme au sein d'une entreprise privée, cette façon de travailler permet une préservation des savoir-faire. Cette méthodologie peut aussi être adoptée avec grand profit lors de la préparation de cours, de didacticiels ou d'ouvrages techniques.

Reproductibilité, replicabilité, répétabilité ?

Le lecteur curieux effectuant une recherche sur Internet s'apercevra vite qu'un débat assez intense porte en ce moment sur la terminologie : on parle beaucoup de reproductibilité (*reproducibility*), de réplicabilité (*replicability*), mais aussi de répétabilité (*repeatability*) [5]. Mon point de vue — fortement influencé par mon passé d'expérimentateur qui continue comme analyste de données et modélisateur (en neurophysiologie) — est que tout paramètre estimé à partir de données expérimentales collectées sur un échantillon d'individus loins d'être tous identiques doit être accompagné d'un intervalle de confiance — sinon, il n'y a pas de « résultat ». Si des collègues refont l'expérience sur une population identique et mesurent le même paramètre, les deux intervalles de confiance doivent se recouvrir ; dans le cas contraire, le résultat n'est pas reproductible [1]. Pour avoir un sens, ces intervalles de confiance doivent prendre en compte la variabilité des individus au sein de la population, les propriétés des instruments de mesure (bruit, erreur systématique, etc), les propriétés des algorithmes d'estimation — ce qui occupe traditionnellement les statisticiens — la précision des flottants utilisée dans les programmes mettant en œuvre les algorithmes, les compilateurs employés, voir même le matériel (CPU/GPU) utilisé pour effectuer concrètement les calculs.

Plan de l'article

La première partie présente, dans un contexte historique, la première « tentative aboutie » de mise en œuvre de la recherche reproductible et ses outils. Un article sur le sujet écrit il y a dix ans se serait arrêter là [2]. La seconde partie présente les développements récents sous l'influence de trois facteurs : la demande croissante d'outils de recherche reproductible de la part de communautés scientifiques — en Biologie et Sciences Humaines notamment — au bagage informatique et numérique « faible » ; conséquence du point précédent, l'usage de plus en plus fréquent de langages interprétés nécessitant de nombreuses bibliothèques — `Python` est ici le principal acteur — dont l'évolution est très (trop) rapide ; le fait que la recherche reproductible exposée en première partie demande un surcroît de travail pour peu de bénéfices *à court terme*, ce qui constitue un problème majeur pour la carrière des (jeunes) chercheurs. Il sera alors temps de conclure cet article.

2 Première époque

Au début de la conclusion d'un article [6] de 1976 intitulé : *Molecular Dynamics and Monte Carlo Calculations in Statistical Mechanics*, Wood et Erpenbeck écrivent :

> Nous souhaitons néanmoins insister sur le fait que ces études [basées sur des simulations] ont de nombreuses caractéristiques communes avec les travaux expérimentaux habituels : elles sont sujettes à des erreurs aussi bien statistiques que systématiques. De ce point de vue, nous considérons que nos ar-

1. Si les deux intervalles sont « à 95 % », la probabilité pour qu'ils ne se recouvrent pas par chance est de $(1-0,95)^2 = 2,5 \times 10^{-3}$.

> ticles devraient respecter les mêmes critères que les articles expérimentaux. Ils devraient ainsi inclure une estimation de l'erreur statistique, une description des conditions expérimentales (c.-à-d. les paramètres des calculs) ainsi que des informations sur la conception de l'appareil de mesure (le programme), une comparaison avec les résultats d'études antérieures, une discussion des erreurs systématiques, etc. C'est seulement dans ces conditions que les résultats pourront être utilisés pour améliorer notre compréhension théorique...

Cette citation traduit, à mon sens, les problèmes entraînés par l'absence de section « Méthodes » dans la plupart des publications que nous classerions aujourd'hui comme « computationnelles ». Mais il se trouve que pour ce type de travail, la section « Méthodes » des articles expérimentaux peut être rendue beaucoup plus explicite grâce aux outils de développement logiciels. C'est ce qu'a démontré l'approche proposée au début des années 90 par le *Stanford Exploration Project*.

Le *Stanford Exploration Project*

En 1992, Jon Claerbout et Martin Karrenbach dans une communication au congrès de la *Society of Exploration Geophysics* écrivent :

> Une révolution dans la formation et dans le transfert technologique résulte du mariage du traitement de texte et des interpréteurs en ligne de commande de type script. Ce mariage permet à un auteur d'associer à chaque légende de figure une étiquette référençant tout ce qui est nécessaire à le régénération de la figure : les données, les paramètres et les programmes. Ceci fournit une exemple concret de reproductibilité en science computationnelle. Notre expérience, au Stanford Exploration Project, montre que la préparation de ce type de document électronique ne demande pas beaucoup plus de travail que celui nécessaire à la préparation d'un rapport classique ; il faut juste tout archiver de façon systématique.

Les outils du *Stanford Exploration Project*

Les géophysiciens du SEP effectuent l'analyse de gros jeux de données ainsi que des simulations de modèles géophysiques « compliqués » (basés sur des EDPs) ; ainsi :

— ils ont l'habitude des langages compilés *et normalisés* comme le `Fortran` et le `C` ;
— ils emploient des moteurs de production comme `Make` ;
— ils écrivent leurs articles en TEX [3] et LATEX [4].

L'idée clé est d'utiliser le moteur de production, non seulement pour générer les « exécutables », mais aussi pour les appliquer aux données grâce à des scripts — et ainsi (ré)générer les figures et les tables de l'article automatiquement —, avant de compiler le fichier `.tex`.

Avec cette approche, nous voyons que *tout* (données, codes sources, scripts, textes) est conservé dans une collection de répertoires imbriqués ce qui rend le travail « facile » à sauvegarder et à distribuer ; de plus un accent est mis dès le départ sur l'utilisation de logiciels libres avec un recours aux langages de programmation normalisés. Par contre, l'emploi de

TEX (ou LATEX) se prête mal à la « prise de notes » et constitue souvent un véritable obstacle pour les chercheurs hors des maths et de la physique. De plus, la gestion d'une arborisation de fichiers, pour ne pas dire l'ensemble de l'approche, s'avèrent « compliquées » dans le cadre d'une analyse exploratoire « au quotidien ».

Les « nouveaux » outils qui permettent à tout un chacun de mettre en œuvre l'approche du SEP

Au début des années 2000, le verrou constitué par LATEX a été essentiellement éliminé par le développement des langages de balisage léger comme : `Markdown` ; `reStructuredText` ; `Asciidoc` ; `Org mode` (utilisé pour écrire ce texte). Avec le logiciel pandoc il est possible de passer quasi instantanément de l'un à l'autre et, grâce à l'extension pandoc de Markdown, un débutant avec une heure de pratique peut générer un fichier LATEX qui ferait envie à un expert [1].

Le mariage du langage de balisage léger avec des langages interprétés populaires comme `Python` et `R` a résulté dans des « cahiers de notes numériques »[2] (*notebooks*) qui permettent très facilement de mélanger un texte descriptif / explicatif avec des lignes de codes. On obtient ainsi des *documents dynamiques* dans lesquels, les figures et les tables ont été remplacées par les instructions qui les génèrent. Il est alors possible de recalculer les résultats, mais aussi d'inspecter et de modifier les instructions qui les ont générés.

La gestion d'arborisation de fichiers est maintenant grandement facilitée par l'apparition du logiciel de gestion de version décentralisé `git` et des serveurs associés `GitHub` et `GitLab`[3]. En fait, la combinaison langage de balisage léger / serveur `git` est tellement efficace et accessible qu'elle peut être employée avec profit par toute personne travaillant sur des textes (je pense aux disciplines littéraires).

Le partage des données, nécessaires à une mise en œuvre complète de la recherche reproductible, a aussi longtemps été un obstacle majeur pour les disciplines qui en génèrent en grande quantité. Les astrophysiciens confrontés très tôt au problème avaient dès les années 70 développés des formats de fichiers permettant de stocker des grandes quantités de données *hétérogènes* — le format `FITS` (*Flexible Image Transport System*), toujours utilisé — ainsi que des serveurs hébergés, par exemple, par la NASA. Ces idées ce sont généralisées avec, au niveau du format de fichiers, le *Hierarchical Data Format* (`HDF5`) et, au niveau des serveurs de données, des initiatives comme zenodo.

Nous disposons ainsi à présent d'outils de bases, essentiellement dérivés du développement logiciel, qui permettent après un apprentissage peu « chronophage » de mettre en œuvre la recherche reproductible. J'ai néanmoins bien conscience que le tour d'horizon proposé dans cette section a été très (trop) bref et j'invite le lecteur curieux d'en savoir plus à suivre le CLOM / MOOC[4] gratuit que nous avons préparé avec mes collègues Arnaud Le-

2. Il s'agit pour le logiciel `R` du paquet `RMarkdown` dont l'usage est rendu aussi simple que celui d'un éditeur de texte par l'environnement de développement RStudio. Les utilisateurs de `Python` peuvent quand à eux employer `Pweave` ou le « carnet de notes » (*notebook*) de `jupyter`.

3. De nombreux instituts de recherche comme l'INRIA, l'INSMI (les maths du CNRS) et de plus en plus d'universités proposent aux chercheurs leurs propres serveurs basés sur `GitLab`

4. Cours en Ligne Ouvert Massif / Massive Open On-line Course.

grand et Konrad Hinsen : Recherche reproductible : principes méthodologiques pour une science transparente.

3 L'évolution actuelle

Nouvelles communautés

Avec la généralisation des approches quantitatives, de plus en plus de communautés scientifiques comme la Biologie au sens large et les Sciences Humaines, produisent une partie de leur résultats par analyse ou simulations sur ordinateurs. Les membres de ces communautés, de part le cursus qu'ils ont suivis, ont rarement une culture informatique / numérique comparable à celle des chercheurs du SEP. Leur bagage dans ce domaine se limite souvent à un cours d'introduction à `Python` (parfois à `Matlab` ou `R`) et ni les langages compilés, ni les moteurs de production comme `Make` ne font partie de leur « boîte à outils ». Leur demande d'outils « tout en un » génère dès lors une profusion de bibliothèques / modules utilisables directement depuis leur langage interprété et interactif favori : `Python` dans la majorité des cas. Tout comme les domaines dans lesquels le besoin de recherche reproductible s'est historiquement fait sentir, les chercheurs de ces « nouvelles communautés » souhaitent eux aussi rendre leur production reproductible. De plus, du fait de la complexité des tâches de récupération et d'agrégation de données ou du fait de la complexité des chaînes de traitement utilisées, ces communautés ont poussé de façon très intéressante le développement d'extension du moteur de production traditionnel pour aboutir aux *workflows* ou *pipelines* modernes — Ricardo Wurmus a donné une présentation courte et lumineuse des enjeux et problèmes des *workflows* dans le cadre de la recherche reproductible au FOSDEM 2021 [5]. Ces outils [6] intéressent *a priori* toute personne impliquée dans la recherche reproductible.

Reproductibilité dans la durée

Une expérience souvent désagréable attend le chercheur qui se lance dans la recherche reproductible : malgré un document dynamique préparé avec le plus grand soin et permettant effectivement de régénérer une étude complète *au moment de la création du document*, cette régénération échoue six mois ou deux ans après. Cet échec résulte de la non prise en compte de la dépendance des résultats (numériques) de l'étude vis à vis de l'environnement logiciel dans lequel celle-ci a été effectuée[5]. Cela peut se traduire concrètement (c'est une expérience tout à fait réelle) par une mise en œuvre de la méthode d'optimisation quasi-Newton BFGS (*Broyden-Fletcher-Goldfarb-Shanno*) ayant complètement changé lors d'une mise à jour de la bibliothèque SciPy de l'« écosystème Python scientifique ». Ainsi, la même optimisation sur les mêmes données ne donne plus les mêmes résultats. Clairement, dans cette situation, une saine attitude scientifique consiste à faire une comparai-

5. https://fosdem.org/2021/schedule/event/guix_workflow/.

6. Il y en a profusion — voir le dépôt GitHub : Awesome Pipeline —, l'évolution est rapide ce qui rend toute recommandation délicate à ce stade.

son détaillée des deux mises en œuvre pour déterminer celle qui est la plus correcte des deux. Mais une tendance lourde en recherche reproductible, la reproductibilité bit à bit[7], « évacue » cette question pour privilégier un critère évaluable par une machine. Des échecs de reproduction dûs à un changement de la mise en œuvre de la méthode BFGS ou à un changement des couleurs par défauts utilisées par la bibliothèque graphique de `Python`, `matplotlib`, se voient ainsi attribués le même poids[8]. Le choix d'une reproductibilité bit à bit combiné à l'adoption de logiciels de « haut niveau », qui évoluent vite, sans que les développeurs se soucient outre mesure de la rétrocompatibilité (*backward compatibility*) de leurs bibliothèques — un problème récurrent de l'écosystème `Python`[9] — expliquent le recours de plus en plus systématique aux conteneurs comme Docker ou Singularity. qui permettent, théoriquement au moins, de figer tous les programmes et logiciels dont dépend une application donnée. Une critique claire de cette approche est développée par R. Wurmus[5]; le lecteur qui voudrait se familiariser rapidement à ce type d'outils pourra consulter avec profit le cours en ligne Reproducible research du projet Code Refinery. Il est clair que la reproductibilité bit à bit constitue le critère idéal de reproductibilité dans une période où les instances d'évaluation se focalisent d'une part sur des indices bibliométriques et, d'autre part, commencent à demander une « recherche reproductible » : avec la reproductibilité bit à bit on fait une telle recherche sans trop perdre de temps — car on fige sa pile logiciel plutôt que de décrire précisément ce qu'on fait — et on produit plus (d'articles).

Il y a potentiellement une alternative que les chercheurs plus âgés, ayant déjà un emploi permanent, peuvent s'offrir le luxe d'explorer : cloisonner au maximum les étapes d'une étude; avoir recours le plus possible à des programmes écrits dans un langage compilé et normalisé comme le `Fortran`, le `C` ou le `C++`; utiliser `Python` au minimum et essayer de se limiter à sa bibliothèque standard. Les langages normalisés évoluent, mais moins vite; une grande importance est accordé par les comités de normalisation à la rétrocompatibilité d'une version à l'autre; plusieurs compilateurs de grande qualité sont systématiquement disponibles. Cette approche n'élimine pas tous les problèmes de dépendance, mais elle les limite fortement.

4 Conclusions

Mettre en œuvre une recherche reproductible « au quotidien » ne présente plus aujourd'hui de gros problèmes. Le chercheur n'a pas besoin de changer de façon radicale sa façon de travailler, juste de systématiser un peu son travail; les outils nécessaires sont disponibles et maintenant bien documentés; des cours sont développés. Mais la recherche reproduc-

7. Une étude génère une collection de fichiers qui peut être vue comme une séquence de bits et l'étude est reproductible si, lorsque le document dynamique est relancé, une séquence identique est obtenue.

8. Dans les cas extrêmes, on accorde plus de crédit à un résultat faux mais reproductible bit à bit, qu'à un résultat scientifiquement correct mais non reproductible bit à bit car la couleur par défaut des graphes est passée du noire au bleu.

9. Le lecteur, pour s'en convaincre, est fortement encouragé à lire la justification de la refonte totale de la génération de nombres (pseudo)aléatoires dans la bibliothèque `numpy`, par le seul (!) développeur responsable de cette fonctionnalité : `https://numpy.org/neps/nep-0019-rng-policy.html`.

tible est une approche jeune qui doit faire face à des problèmes pas toujours pleinement anticipés, comme la reproductibilité dans la durée. Comme toute discipline nouvelle et dynamique elle voit se présenter de nombreuses propositions de solutions, pas toujours compatibles, au problème rencontrés. Nous avons ainsi aujourd'hui de nombreux moteurs de *workflow* à disposition, plusieurs systèmes de conteneurs, etc. Comme l'un des enjeux majeurs est la fiabilité dans le temps il va nous falloir nécessairement faire preuve de patience et rester ouverts.

Références

[1] Jean-Daniel BONJOUR. *Élaboration et conversion de documents avec Markdown et Pandoc*. Sous licence CC BY-SA 3.0. EPFL-ENAC-IT, sous licence CC BY-SA 3.0. 2014. URL : http://enacit1.epfl.ch/markdown-pandoc/.

[2] Matthieu DELESCLUSE et al. "Making neurophysiological data analysis reproducible : Why and how?" In : *Journal of Physiology-Paris* 106.3-4 (mai 2012), p. 159-170. URL : http://dx.doi.org/10.1016/j.jphysparis.2011.09.011.

[3] Donald E. KNUTH. *The TeXbook*. Reading, Massachusetts : Addison-Wesley, 1984, p. x+483.

[4] Leslie LAMPORT. *LaTeX : A Document Preparation System*. Reading, Massachusetts : Addison-Wesley, 1986.

[5] NATIONAL ACADEMIES OF SCIENCES, ENGINEERING AND MEDICINE. *Reproducibility and Replicability in Science*. Washington, DC : The National Academies Press, 2019. URL : https://www.nap.edu/catalog/25303/reproducibility-and-replicability-in-science.

[6] W W WOOD et J J ERPENBECK. "Molecular Dynamics and Monte Carlo Calculations in Statistical Mechanics". In : *Annual Review of Physical Chemistry* 27.1 (oct. 1976), p. 319-348. URL : http://dx.doi.org/10.1146/annurev.pc.27.100176.001535.