Notebook and Open science : toward more FAIR play

Mariannig Le Béchec, Célya Gruson-Daniel, Clémence Lascombes, Émilien Schultz

Notebook and Open science : toward more FAIR play

Mariannig Le Béchec[1,2*], Célya Gruson-Daniel[3,4] , Clémence Lascombes[3] , Émilien Schultz[5]

1 ELICO, Université Claude Bernard Lyon 1

2 URFIST de Lyon - Université Claude Bernard Lyon 1

3 Inno[3]

4 COSTECH – Université de Technologie de Compiègne

5 CREST – ENSAE Paris

*Contact address : mariannig.le-bechec@univ-lyon1.fr

Abstract

Notebooks are now commonly used in digital research practices. Despite their increasing ubiquity, the characteristics, roles, and uses associated with notebooks have been seldom studied from a science and technology studies (STS) perspective. In this article, we present an overview of the empirical work on notebooks in order to describe existing practices, typologies crafted to grasp their diversity, and their limitations when used in data analysis workflows. Following this review, which highlights a focus of studies on data science rather than research practices in academic contexts, we discuss the role of notebooks as a vector and lever for the FAIR[1] (Findable, Accessible, Interoperable, Reusable) principles associated with open science.

Keywords

notebook, literate programming, jupyter, open science, FAIR

INTRODUCTION

In digital research practices, researchers commonly adopt notebooks as a media that allows the joint integration of content elements (particularly textual), programming elements (code produced in different languages), and the results of these treatments once executed. Their diversity is notable: some are documents that are first written and then compiled to obtain results, such as those made possible by Rmarkdown[2], or directly interactive documents such as those of the Jupyter[3] project, subsequently referred to as "computational notebooks"[4].

---

[1] https://force11.org/info/the-fair-data-principles/

[2] https://rmarkdown.rstudio.com/

[3] https://jupyter.org/

[4] "A computational notebook consists of a sequence of cells, each containing code, prose, or media generated by the notebook's computations (e.g., graphs), embodying a combination of literate programming and read-eval-print-loop (REPL) interaction." (Liu et al., 2023)

These documents differ from other types of digital documents, which are also referred to as notebooks, such as electronic lab notebooks. The scientific literature[5], on electronic lab notebooks, addresses them separately, and they are not discussed here.

Notebooks are now a common tool in teaching programming and data processing. Despite this adoption, there has been a lack of research into their characteristics, roles, and uses in the field of science and technology studies (STS). This article provides a comprehensive review of extant empirical studies concerning notebooks, delineating current practices, cataloguing the variety of notebook formats, elucidating criticisms, and proposing enhancements.

We searched scientific databases (refer to the "Methodology of the literature review" section for details) selecting empirical work. Emphasis was placed on the identification of open science practices and the application of legal principles to notebooks[6].

Noticing that existing literature focuses more on the field of data science than research practices themselves, we then discuss the role of notebooks in promoting the FAIR (Findable, Accessible, Interoperable, Reusable) principles of open science. We foresee several paths for future research with the NOOS (Notebook for Open Science) project.

I. METHODOLOGY OF THE LITERATURE REVIEW

Given the limited published literature on this emerging topic, our primary aim was to establish the current state of knowledge on notebooks (2023). To achieve this, we conducted a non-systematic 'scoping review', iteratively focusing on the keywords in the field.

Our research concentrated on empirical work related to computational notebooks. We queried many scientific databases, including Dimensions, BASE Api, and Semantic Scholar, as well as the Zotero collection of the Notebook Working Group[7], which was founded in 2021. We used the following keywords and combinations: '*notebook*' OR '*interactive notebook*' OR '*computational notebook*' OR '*executable notebook*' OR '*jupyter notebook*' AND '*open science'*.

The four authors conducted a systematic reading of the articles to only take into account empirical contributions to the topic. Any discussions lacking investigation or personal viewpoint were excluded. At the end, we selected 12 articles for a comparative study of the methodologies used, major results, disciplines, and types of notebook (see appendix table).

II. RESULTS

2.1 An emerging literature focusing on software engineering and data science

[5] See : Rabemanantsoa, Tovo, Dominique Pigeon, Nicolas Gilles Mathieu, Christophe Chipeaux, Simon Duvillard, Célya Gruson-Daniel, Marie Herbet, et al. 2021. « Report of the Working Group on Electronic Lab Notebooks ». Report, Comité pour la science ouverte. https://doi.org/10.52949/30.

[6] This preliminary work outlines the methodology that will be used in the NOOS (Notebook for Open Science) study, which was the winner of the GIS Réseau URFIST 2022 call for projects. It clarifies the issues that will be addressed in this research project.

[7] https://www.zotero.org/groups/4416056/gt-notebooks/items/PGT9T72T/library

The literature dedicated to notebooks is recent and addresses primarily computer and software engineering, as well as human-software interaction. IT engineers, data scientists in the private sector, and academics alike use these software environments developed by IT engineers. The distinction between industrial and scientific purposes can be blurred, making it challenging to summarise practices.The studies often took a solutionist approach. Technological solutions were proposed to resolve the identified problems. By having built-in, notebook-specific testing and linting frameworks, as well as features for code refactoring and modularization, data scientists can write high-quality notebooks without compromising on timeliness (Quaranta et al., 2022). The articles highlighted solutions that are designated or refer to already existing applications (Chattopadhyay et al., 2020; Kery et al., 2018).

Issues related to academic research were sometimes addressed, especially when contributors originate from an academic institution (Quaranta et al., 2022). However, in the case of interviews with academic actors, disciplinary specificities are completely absent (Quaranta et al., 2022; Rule et al., 2018). In the literature on scientific programming, a well-identified tension between researchers and software engineers was reported: "moreover, many notebook authors identify as scientists (e.g., chemists), and so may not have been exposed to concepts and skills related to reducing technical debt [...] and maintaining conceptual integrity, such as refactoring and software design principles" (Liu et al., 2023). The article only addressed a few specific issues, such as the dissemination of results. However, some articles focus on scientific research but fail to emphasise the specificities of the field (Samuel & Mietchen, 2022; Wagemann et al., 2022).

The articles focused on programming practices, reporting discussion on best practices in software engineering, and limitations of software environments associated with notebooks. Questions related to the integration of notebooks in a more general perspective of professional activities, particularly in scientific research, are rarely addressed, except for Samuel & Mietchen (2022). Yet, notebooks in laboratories raised many questions. For instance, is it possible to categorise practices through disciplines or user status? Additionally, the articles did not address the factors that contribute to the spread of practices or the limitations of adoption. For instance, there was no data on the percentage of users based on their professional field or discipline, or on alternative applications. It is important to note that most of the studies are prescriptive and focus on best practices : "all of these studies suggest that disciplined and informed use of notebooks, guided by shared best practices, is essential to successfully support data science work" (Quaranta et al., 2022).

In terms of study methodology, the primary focus of these studies was on analysing digital traces of notebooks. These notebooks were extracted from GitHub, a collaborative code version management platform, and Kaggle[8], a data science machine learning competition platform, both widely used in the IT industry. This approach allowed to provide a comprehensive analysis of the structure of notebooks (Pimentel et al., 2021), their location within repositories (Rule et al., 2018), and their history, particularly in relation to commits[9] (Raghunandan et al., 2023). The availability of digital data thus enables the definition of various precise quality indicators for the notebooks, such as structural or stylistic code metrics (e.g., the number of code comments, functions, etc.) (Grotov et al., 2022). However, the representativeness of those samples is questionable. The authors investigated both the contexts in which notebooks are created and the users themselves to a limited extent, relying

[8] Kaggle offers a system of medals on notebooks and ranks for individuals: Novice, Contributor, Expert, Master and Grandmaster (Choetkiertikul et al. 2023).

[9] Save changes in the source repository: https://git-scm.com/docs/git-commit/fr

on information available on their profiles (Liu et al., 2023) or via the profession obtained in interviews. Besides, it seems that although notebook users have different profiles they have mostly the same uses of this tool.

Several studies not only analysed digital data but also included semi-structured interviews with notebook users. While these interviews investigated practices and judgements made about notebooks, their focus remained confined to the notebook itself, neglecting the broader work context and associated activities in which the notebook finds application. Moreover, the aspects of socialisation to the tools and training were not considered in these interviews. Only one study mentioned administering a questionnaire (Chattopadhyay et al., 2020). In research aimed at prototyping new tools, UX test methods were implemented. However, these methodologies rarely mobilise observations in context, except for Chattopadhyay's work (ibid). Furthermore, no study seemed to investigate notebooks within the broader questions of archives and their articulation with other associated information, such as data, metadata, licences, and README files.

2.2 The Rise of Jupyter Notebooks

Overall, it appears that the literature primarily focused on computational notebooks, particularly those from the Jupyter project, which have been extensively addressed. While other notebook solutions, such as RStudio, Sage, and their platform versions (Microsoft Azure, Google Colab, Kaggle, Databricks, Apache Zeppelin), are mentioned, they are less common.

Although several studies (Pimentel et al., 2021; Samuel & Mietchen, 2022) highlighted the importance of reproducibility, the literature put the emphasis on the lack of thereof especially on GitHub (cf. Identification of numerous limitations and criticisms). However, this criticism was not specific to notebooks and applies to code (Trisovic et al., 2022)[10]. According to research, several factors influence reproducibility, such as notebook popularity. Researchers identified several obstacles to reproducibility, such as the non-linear organisation of notebooks and the absence of documentation regarding the libraries used and their provenance (Ramasamy et al., 2023; J. Wang et al., 2020; Zhang et al., 2020). Moreover, the presence of computational notebooks adds complexity to the concept of reproducibility. Traditionally, reproducibility referred only to the absence of code or data (Baker, 2016) or to disciplinary distinctions (Leonelli, 2018). Nonetheless, in the present day, these definitions must also incorporate the suitable documentation for code and data availability.

Studies suggested that notebooks provide a space where the limitations of systematic programming practices and established best practices, which are normally enforced in the code, are loosened. Despite the knowledge of best coding practices, notebooks offer greater freedom and a constant tension between quality and speed of development (Quaranta et al., 2022). Notebooks seemed to be characterised by a specific programming style (Grotov et al., 2022) and introduced new ways of documenting code (A. Y. Wang et al., 2021). The potential decline in quality may arise from the use of notebooks by multi-disciplinary teams, presenting

[10] Trisovic et al. (2022) highlight that over 74% of R files are unable to be executed without error due to poor code quality. “We find that 74% of R files failed to complete without error in the initial execution, while 56% failed when code cleaning was applied, showing that many errors can be prevented with best coding practices.”

challenges to ensuring consistent programming practices and documentation (A. Y. Wang et al., 2021).

Studies of large samples of notebooks, particularly those available on GitHub (Rule et al., 2018), showed their diversity and made it possible to establish classifications. Notebooks have diverse uses, such as teaching or demonstrating materials, as a step in data analysis, or as a training material. Pedagogical uses appeared important (Rule et al., 2018): out of 69 random notebooks, “31 notebooks associated with courses, such as tutorials, class assignments, or course exercises” (Pimentel et al., 2021). The categorization of Liu et al. 2021 proposed five types of notebooks:

- *Exploratory Analysis*, which represents the majority of uses;
- *Programming Assignment*;
- *Technology Demonstration*);
- *Analytical Demonstration*;
- *Educational Material*.

The non-linear and evolving nature of this type of document was widely emphasised (Ramasamy et al., 2023). There are two distinct poles of practice: the exploratory pole and the explanatory pole. On one hand, the exploratory pole brings together attempts, sketches, and pieces of code and results, which are far from the best practices of software engineering. On the other hand, the explanatory pole consists of more finalised documents that meet stricter constraints of narrative (exposition of the approach and sequence of treatments) and reproducibility. “These studies demonstrate a tension between exploration and explanation in constructing and sharing computational notebooks” (Rule et al., 2018). The most recent work focused precisely on the evolving nature of notebooks, which can develop a specific trajectory according to their uses (Liu et al., 2023; Raghunandan et al., 2023). It is therefore essential to consider the computational notebooks produced as heterogeneous entities and move towards more detailed classification work.

2.3 Recommended best practices

The literature frequently addresses the question of 'best practices', which can refer to effective methods or a standard of quality. There is significant overlap between articles on this topic, resulting in a comprehensive understanding. In this particular context, there was a significant overlap between the articles, suggesting that a comprehensive coverage has been accomplished. “It is worth noting that none of the interviewees mentioned best practices that had not been already identified through the literature review, thus increasing the confidence that we may have reached theoretical saturation, given the current state of the art and practice” (Quaranta et al., 2022).

The cross-sectional observation is that these ‘best practices’ were known, but mainly not applied (Quaranta et al., 2022). Pimentel et al. (2021) provided eight recommendations for notebook reproducibility, which can be summarised in a table.

General categories	Best practices
Make your analysis traceable and reproducible	Use a version control system to manage project dependency. Managing project dependencies Providing applications without third-party dependencies Put imports at the beginning of the file

	Ensure that the entire code functions correctly, not just the modified part
Write quality code (i.e. code that can be easily shared and reused)	Structure your code into modules (abstract the code into functions and place them in a dedicated module; place dependencies at the beginning of the notebook) Test your code Name your notebooks consistently Respect standards Use relative paths Define requirements
Exploiting the paradigm of literate programming	Document your code for yourself and others Use Markdown headings to structure your notebook
Keeping your notebook clear and concise	Keeping your notebook clear Keeping your notebook concise
Differentiate between artefacts produced during development and production	Differentiate between artefacts produced during development and production
Adopting open distribution	Making your notebook available Making your data available

Table 1: Catalogue of good practices specific to the use of notebooks extracted from the literature review

Besides discussion regarding best practices, there was also the matter of the integration of the notebook into a wider environment. Notebooks rely on a broader software environment. Regarding distribution, authors recommend including a README file, an open-source licence for reuse, and both static and dynamic versions of the notebooks that do not require local installation of Jupyter to be read or executed (Quaranta et al., 2022). However, the concept of open science and related work were not addressed.

2.4 Identification of numerous limitations and criticisms

Notebooks were often praised for their educational value, whether for presenting courses (Wagemann et al., 2022) or exercises and tutorials (Pimentel et al., 2021). However, they have also been subject to numerous criticisms, from downloading data to disseminating or collaborating in a notebook (Chattopadhyay et al., 2020).

These challenges hinder the attainment of the desired outcomes for notebooks, such as reproducibility and code narration, in a literate programming environment (Kery et al., 2018). As noted by Wangeman et al. (2022), there is a common misconception that content is automatically reproducible when presented as a Jupyter notebook. Researchers often made proposals for improving or remedying the shortcomings of notebooks in response to criticisms (Quaranta et al., 2022; Ramasamy et al., 2023; Rule et al., 2018).

Academic research practices appeared to be often hindered by poor reproducibility, as studies analysing many notebooks have evidenced. Pimentel et al. (2021) found that only 14% of the notebooks they studied were reproducible.

Several reasons are given throughout the articles:

- Lack of documentation. Rule et al. (2018) report that more than a quarter of the notebooks studied contain no documentation. Even if the notebooks were used for exploration rather than to explain and narrate results, several essential pieces of documentation were missing, such as the list of third-party software used (Ramasamy et al., 2023; A. Y. Wang et al., 2021) or the origin of the code used (Zhang et al., 2020). The collaborative approach and numerous iterations would amplify this phenomenon throughout the life cycle of the notebook (A. Y. Wang et al., 2021). Regarding reproducibility and reuse, it was noted that "Unfortunately, even seemingly simple reuse can become more complicated than expected, such as when the "earlier notebook uses absolute paths" (IP5), when the "cells have no designated format or function" (IP1) and can't be easily isolated, and when there are complex dependencies to bring into the new notebook" (Chattopadhyay et al., 2020);

- The lack of consistency in cell execution (Guzharina & Guzharina, 2020). The non-linear structure of the notebook and the ability to execute specific parts of it without running the entire notebook from top to bottom can make it difficult to ensure transparency in the execution mechanisms;
- Version tracking has been a constant challenge for Jupyter users since 2015. A UX questionnaire proposed by Jupyter highlighted the difficulty of not being able to version notebooks, which often results in a proliferation of files without the ability to track changes over time (Kery et al., 2018);
- The quality of the code was frequently low (A. Y. Wang et al., 2021). Copying and pasting between several notebooks and duplicating notebooks (Quaranta et al., 2022) contribute to the dissemination of low-quality code. The lower quality of the code was suspected to be connected to the very use of the notebook as an exploratory tool rather than for demonstrating or presenting results. Speed was favoured to quality, and this is close to the idea of the 'messy notebook' that emerged in several articles.

Furthermore, the notebook format received criticism due to the constraints associated with its interface, which can result in errors or lower quality code. For instance, some articles noted that notebooks become excessively heavy beyond a certain length, which can lead to crashes (Kery et al., 2018). One criticism of working with massive data is the difficulty of navigating through all the documents due to the complex and linear structure of the code. It can be also challenging to visualise the workflow stages (Ramasamy et al., 2023). In addition, the format of the cells or of the visualisation renderings can make it difficult to use the notebook or to disseminate the results effectively, leading to results that are difficult to appropriate (Chattopadhyay et al., 2020).

Several solutions have been proposed to address these areas of friction. These solutions included facilitating the security of confidential data and preserving modification histories, both of which are considered major challenges. However, there was seldom mention of how to improve notebooks from an open science perspective (availability of data, conditions for re-use, use of licences) (see Discussion: Are notebooks a strategic tool for open science?).

III. DISCUSSION: ARE NOTEBOOKS A STRATEGIC TOOL FOR OPEN SCIENCE?

Building on this literature review, we focus now on the relationship between notebooks and the principles of open science. Open science aims to promote the dissemination of knowledge as soon as it is available, using digital and collaborative technologies[11]. Articles, data, and scripts (source codes) are the main elements targeted for sharing and openness. Open access

[11] An approach to the scientific process that focuses on spreading knowledge as soon as it is available using digital and collaborative technology.
https://research-and-innovation.ec.europa.eu/strategy/strategy-2020-2024/our-digital-future/open-science_en

has long been at the heart of open science, and it is worth highlighting that open science lays at the heart of the Jupyter project (Schultz, 2023). Nevertheless, source codes and software have only recently been the subject of public policy attention in open science, although the open source and free software movements are largely rooted in organisational, economic, and legal principles.

We propose three main points for seizing computational notebooks from an open science perspective: a better understanding of the practices of research professionals, explicit consideration of the issues surrounding the FAIRisation of notebooks, and the conceptualisation of notebooks in a wider environment of free circulation of scientific codes and productions, in particular through open licences.

3.1 What are research professionals doing with notebooks?

Existing studies focused on the broad field of data science, with no specific interest in the academic world (research within public organisations, research institutes, universities, etc.). Therefore, there is a substantial deficiency in terms of specific knowledge about practices in these environments. However, studying the uses of these devices is necessary to gain a better understanding of their applications and the specificities that may exist within the scientific world itself, taking into account the differences between disciplines, statuses, and research approaches. According to several studies, there is variability in the digital practices of research professionals[12]. Although there were testimonials and feedback (Beg et al., 2021), there is minor work dedicated to the practices of research professionals in relation to notebooks: the only systematic article dealt with the biomedical field (Samuel & Mietchen, 2022).

More specifically, notebooks were mainly presented as a new interface between source code and publication via executable articles. The aim is to get back to the processing of results and ensure transparency and reproducibility. In 2018, an article in The Atlantic was titled 'The scientific paper is obsolete' [13]. However, research on computational notebooks, particularly the Jupyter project, shows that they are mainly used for exploratory purposes, and only a small proportion of them are destined to become fully-fledged, published media (see Putting notebooks in their proper place in the ecosystem). Additionally, notebooks serve different purposes, ranging from recording exercise results to prototyping intermediate stages of applications. Therefore, it is important to identify the various situations in which these notebooks are used. Even if teaching seems to play an influential role, the role of notebooks in collaborative dynamics needs to be explored, especially as an interface between the scientific world carrying out the processing and “non-specialists”, and as a broker in scientific publication practices.

In addition, the current literature did not address the conditions that lead a user to start using a notebook or the interdependence between notebooks and other formats such as programming scripts, software, and forge. Furthermore, there is a lack of data on the profiles of the main users. Therefore, further research is needed to identify the levers and obstacles associated with the spread of this medium in different communities. Many limitations and criticisms have been addressed (see Identification of numerous limitations and criticisms), highlighting the

[12] For example, the survey State of open science practices in France (Béchec et al., 2022) or the study "Décliner la Science Ouverte"(Gruson-Daniel & Groupe Projet Réussir L'Appropriation De La Science Ouverte, 2022).

[13] https://www.theatlantic.com/science/archive/2018/04/the-scientific-paper-is-obsolete/556676/

different needs of different users. Thus, we can expect different levels of adoption. Scientific communities, including those in the medical sciences, who are already familiar with programming tools, may use these new tools differently than communities in more distant disciplines, such as the social sciences.

3.2 Moving towards the FAIRisation of notebooks

Although recommendations on formatting, writing, and tools for improving practices often accompany criticisms directed at notebooks, seldom attention has been given to the dimensions of dissemination and openness. In the articles examined, advice on dissemination remains general (see Table 1). The FAIR principles, or more specifically the use of persistent identifiers such as the DOI, are not mentioned. Currently, there is a lack of practical guidelines to facilitate the application of the principles of open science and FAIR data management in the notebook environment[14].

Most research on notebooks and their improvements focused on data engineering applications. However, the relationship between notebooks and open science, in terms of best practices, was rarely mentioned. Despite the fact that the genesis of projects, such as Jupyter, is directly in line with the open science perspective (Schultz, 2023). Nonetheless, several points of junction between open science and notebooks could be identified:

- Code sharing in scientific articles is often facilitated through the use of notebooks (Wofford et al., 2020). Notebooks are key tools for implementing the principles of literary programming, transforming scientific articles into executable entities which can be read and executed simultaneously;
- Notebooks are also seen as a major pedagogical interface for learning to program in various fields of science (Hanč et al., 2020);
- They offer great flexibility compared to other tools, without imposing complex chains of operations;
- Notebooks facilitate the sharing and organisation of computational analyses through simple editing and writing rules (Rule et al., 2019);
- The open-source and free nature of notebooks makes them particularly suitable for pedagogical contexts and is part of a constantly evolving community, which expands their functionalities.

The issue of notebook reproducibility, as discussed in the literature, is closely linked to the challenges of open science. There are several areas that require further exploration, such as ensuring complete references to sources and data, standardising technical ecosystems, and establishing citation guidelines for notebooks. Extending FAIR principles in notebooks could benefit from being identifiable (findable), for example by considering issues of accessibility and indexing, in particular the identification of notebook versions during archiving, or re-use with mention of free licences.

3.3 Putting notebooks in their proper place in the ecosystem

Over and above the issues specific to scientific research - whether it be the practices of research professionals or the FAIRisation of media - the literature review showed that existing work focuses almost exclusively on notebooks themselves (uses or content). However, the

[14] A FAIR4RS working group has looked at the FAIR process for software, but this does not specifically concern notebooks, which have different uses and purposes (Katz et al., 2021).

extent of their integration into digital ecosystems has not been thoroughly investigated. Two points are worth highlighting.

Firstly, the legal framework surrounding notebooks and their environment is scarcely discussed. In 2019, Schröder et al. demonstrated a significant absence of licences on Jupyter notebooks, with at least a third of resources lacking licences (Schröder et al., 2019). When licences were present, the most common ones were MIT, GPLv and CC0. However, the authors did not provide any justification for their choice of licence. The use of the Creative Commons Zero (CC0) licence is not common in the open-source field, where the main distinctions are between permissive licence (such as the MIT licence) and copyleft licence (such as GPLv). Others use proprietary licences. Besides, there are few or no recommendations for citing the source code or data in notebooks accompanying the publication of articles (Edelmann et al., 2020).

In the cases examined, notebooks are often stored on repositories, notably GitHub, as part of broader research projects. The licence may be outlined in a text appendix file rather than the notebook itself. This raises the issue of integrating the notebook into wider repositories, which include scripts, data, and third-party documents in various formats, along with all the dependencies and software required for execution. To gain a better understanding of the notebook ecosystem, it is necessary to shift the focus towards its infrastructure and practical applications. The articles identified numerous dependencies, such as GitHub and Binder, but it is important to determine if these services are still in use and if there are different sets of ecosystems. This approach is crucial to avoid treating computational notebooks as fixed, autonomous entities.

IV. CONCLUSION: CONTINUE THE INVESTIGATION

The scientific literature on the computational notebooks, which focus mainly on Jupyter's, revealed new practices and opened the discussion on how to stabilise them. Although the genesis of the notebooks (IPython becoming Jupyter) is closely linked to the issue of open source and the openness of scientific knowledge, existing surveys that focus on data science rarely address this topic. Although the 'best practices' identified may be applicable to research practitioners, it is likely that the issues of exploration and scientific publication lead to specific problems.

This article presented a selection of the most representative articles conducting empirical studies on notebooks. It describes the limitations and criticisms raised against notebooks, as well as the 'good practices' proposed and the associated tools. Additionally, it discussed the challenges of notebooks as a lever for appropriating and integrating an open science approach, specifically by mentioning the FAIR principles. Although reproducibility is a crucial aspect emphasised by the use of notebooks in open science discourse, achieving it in practice can be challenging. Therefore, our goal has been to outline several approaches for making notebooks FAIR, which involves considering the various user audiences in academic research (disciplinary, epistemic, practical communities, etc.). However, the notebooks' integration into a research infrastructure and tool ecosystem is necessary to promote the open circulation and exploitation of knowledge.

As part of the NOOS project continuation, we aim to map out the practices and uses of notebooks and their relation to open science issues, such as open access to scientific publications, open data and software, and the appropriation of knowledge by various communities. This will be achieved through the use of focus groups and interviews. The study aims to propose models for FAIR notebooks and analyse a set of notebooks and their environment. The notebooks will be mapped to define the most important open science principles to be integrated based on their types and objectives.

V. REFERENCES AND CITATIONS

5.1 References

Alena Guzharina & Alena Guzharina. (2020, décembre 17). *We Downloaded 10,000,000 Jupyter Notebooks From GitHub – This Is What We Learned*. The JetBrains Blog. https://blog.jetbrains.com/datalore/2020/12/17/we-downloaded-10-000-000-jupyter-notebooks-from-github-this-is-what-we-learned/

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, *533*(7604), Article 7604. https://doi.org/10.1038/533452a

Beg, M., Taka, J., Kluyver, T., Konovalov, A., Ragan-Kelley, M., Thiery, N. M., & Fangohr, H. (2021). Using Jupyter for Reproducible Scientific Workflows. *Computing in Science & Engineering*, *23*(2), 36-46. https://doi.org/10.1109/MCSE.2021.3052101

Chattopadhyay, S., Prasad, I., Henley, A. Z., Sarma, A., & Barik, T. (2020, avril 1). *What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities*. ACM CHI Conference on Human Factors in Computing Systems. https://www.microsoft.com/en-us/research/publication/whats-wrong-with-computational-notebooks/

Edelmann, A., Wolff, T., Montagne, D., & Bail, C. A. (2020). Computational Social Science and Sociology. *Annual review of sociology*, *46*(1), 61-81. https://doi.org/10.1146/annurev-soc-121919-054621

Grotov, K., Titov, S., Sotnikov, V., Golubev, Y., & Bryksin, T. (2022). *A Large-Scale Comparison of Python Code in Jupyter Notebooks and Scripts* (arXiv:2203.16718). arXiv. https://doi.org/10.48550/arXiv.2203.16718

Hanč, J., Štrauch, P., Paňková, E., & Hančová, M. (2020). *Teachers' perception of Jupyter and R Shiny as digital tools for open education and science* (arXiv:2007.11262). arXiv. https://doi.org/10.48550/arXiv.2007.11262

Katz, D. S., Gruenpeter, M., & Honeyman, T. (2021). Taking a fresh look at FAIR for research software. *Patterns*, *2*(3). https://doi.org/10.1016/j.patter.2021.100222

Kery, M. B., Radensky, M., Arya, M., John, B. E., & Myers, B. A. (2018). 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*, 1-11. https://doi.org/10.1145/3173574.3173748

Leonelli, S. (2018). Rethinking Reproducibility as a Criterion for Research Quality. In *Including a Symposium on Mary Morgan : Curiosity, Imagination, and Surprise* (Vol. 36B, p. 129-146). Emerald Publishing Limited. https://doi.org/10.1108/S0743-41542018000036B009

Liu, E. S., Lukes, D. A., & Griswold, W. G. (2023). Refactoring in Computational Notebooks. *ACM Transactions on Software Engineering and Methodology*, *32*(3), 77:1-77:24. https://doi.org/10.1145/3576036

Pimentel, J. F., Murta, L., Braganholo, V., & Freire, J. (2021). Understanding and improving the quality and reproducibility of Jupyter notebooks. *Empirical Software Engineering*, *26*(4), 65. https://doi.org/10.1007/s10664-021-09961-9

Quaranta, L., Calefato, F., & Lanubile, F. (2022). Eliciting Best Practices for Collaboration with Computational Notebooks. *Proceedings of the ACM on Human-Computer Interaction*, *6*(CSCW1), 1-41. https://doi.org/10.1145/3512934

Raghunandan, D., Roy, A., Shi, S., Elmqvist, N., & Battle, L. (2023). Code Code Evolution : Understanding How People Change Data Science Notebooks Over Time. *Conference on Human Factors in Computing Systems - Proceedings*. https://doi.org/10.1145/3544548.3580997

Ramasamy, D., Sarasua, C., Bacchelli, A., & Bernstein, A. (2023). Visualising data science workflows to support third-party notebook comprehension : An empirical study. *Empirical Software Engineering*, *28*(3), 58. https://doi.org/10.1007/s10664-023-10289-9

Rule, A., Birmingham, A., Zuniga, C., Altintas, I., Huang, S.-C., Knight, R., Moshiri, N., Nguyen, M. H., Rosenthal, S. B., Pérez, F., & Rose, P. W. (2019). Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. *PLOS Computational Biology*, *15*(7), e1007007. https://doi.org/10.1371/journal.pcbi.1007007

Rule, A., Tabard, A., & Hollan, J. D. (2018). Exploration and Explanation in Computational Notebooks. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1-12. https://doi.org/10.1145/3173574.3173606

Samuel, S., & Mietchen, D. (2022). *Computational reproducibility of Jupyter notebooks from biomedical publications*. https://doi.org/10.48550/ARXIV.2209.04308

Schröder, M., Krüger, F., & Spors, S. (2019). *Reproducible Research is more than Publishing Research Artefacts : A Systematic Analysis of Jupyter Notebooks from Research Articles* (arXiv:1905.00092). arXiv. https://doi.org/10.48550/arXiv.1905.00092

Schultz, E. (2023). *Du laboratoire à Jupyter : La trajectoire d'un instrument logiciel libre de la science ouverte*. https://hal.science/hal-04316428

Trisovic, A., Lau, M. K., Pasquier, T., & Crosas, M. (2022). A large-scale study on research code quality and execution. *Scientific Data*, *9*(1), Article 1. https://doi.org/10.1038/s41597-022-01143-6
Wagemann, J., Fierli, F., Mantovani, S., Siemen, S., Seeger, B., & Bendix, J. (2022). Five Guiding Principles to Make Jupyter Notebooks Fit for Earth Observation Data Education. *Remote Sensing*, *14*(14). https://doi.org/10.3390/rs14143359
Wang, A. Y., Wang, D., Drozdal, J., Liu, X., Park, S., Oney, S., & Brooks, C. (2021). What Makes a Well-Documented Notebook? A Case Study of Data Scientists' Documentation Practices in Kaggle. *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 1-7. https://doi.org/10.1145/3411763.3451617
Wang, J., Kuo, T., Li, L., & Zeller, A. (2020). Restoring reproducibility of Jupyter notebooks. *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*, 288-289. https://doi.org/10.1145/3377812.3390803
Wofford, M. F., Boscoe, B. M., Borgman, C. L., Pasquetto, I. V., & Golshan, M. S. (2020). Jupyter Notebooks as Discovery Mechanisms for Open Science : Citation Practices in the Astronomy Community. *Computing in Science & Engineering*, *22*(1), 5-15. https://doi.org/10.1109/MCSE.2019.2932067
Zhang, A. X., Muller, M., & Wang, D. (2020). *How do Data Science Workers Collaborate? Roles, Workflows, and Tools* (arXiv:2001.06684). arXiv. http://arxiv.org/abs/2001.06684

5.2 Appendices

Appendix 1 Table of the 12 identified references

Reference	Title	Speciality	Type of *notebooks*	Objective	Method	Main finding	Science ?	Recommendations?	Practices ?
Raghunandan et al., 2023	*Code Code Evolution: Understanding How People Change Data Science Notebooks Over Time*	*Computational engineering*	Jupyter	Examine the development of notebooks in the process of constructing meaning.	2574 notebooks from GitHub	Quantify the type of notebook (exploratory/explanatory)	No	Tooling to identify the position of the notebook on the E/E axis	Indirectly, a scale for stabilisation
Liu et al., 2023	*Refactoring in Computational Notebooks*	*Computational engineering*	Jupyter	Study the evolution of code in public notebooks	200 notebooks with commit on Github	Different practices for different notebook genres and authors	Partially	On improving refactoring tools	Indirectly, the evolution of notebooks
Ramasamy et al., 2023	*Visualising data science workflows to support third-party notebook comprehension: an empirical study*	*Computational engineering*	Jupyter	Developing a strategy for displaying notebooks	470 notebooks and controlled user experimentation with 35 data scientists	Clarification of data science workflow terminology; possibility of improving the use of notebooks by visualising workflows	Partially	Visualisation tool; making workflow more explicit	Yes, notebook reuse
Grotov et al., 2022	*A Large-Scale Comparison of Python Code in Jupyter Notebooks and Scripts*	*Computational engineering*	Jupyter	Examine the coding style used in Jupyter notebooks.	847881 notebooks with open licence	Notebook code has a different style (lower complexity; more interdependency; more style errors)	No	A package (linter) to measure style	No
Samuel et al., 2024	*Computational reproducibility of Jupyter notebooks from biomedical publications*	*Computational engineering*	Jupyter	Testing the reproducibility of notebooks associated with biomedical publications and the gap between recommendations and practices.	9625 notebooks from Pubmed publications	Discusses the concept of reproducibility. Demonstrates the low reproducibility	Yes	Better documentation	No
Wagemann et al., 2022	*Five Guiding Principles to Make Jupyter Notebooks Fit for Earth Observation Data Education*	*Eath Science*	Jupyter	Present the process for creating notebooks for a course and the main principles involved	Creation of 70 notebooks	Identifying principles: the importance of text cells; navigation elements; following the principles of scientific programming; using the Jupyter ecosystem to share; aiming for reproducibility	Yes	Improving the final characteristics of notebooks	Feedback from an experiment
Quaranta et al., 2022	*Eliciting Best Practices for Collaboration with Computational Notebooks*	*Computational engineering*	Jupyter	Identify best practices for data scientists working with notebooks	Systematic literature review; 22 interviews with data scientists; 1,380 Kaggle	List of 17 best practices from the literature (table); Individuals are aware of best practices but do not necessarily apply them	Partially	Improve the notebook environment built-in	Yes, numerous interview transcripts

					notebooks				
Wang et al., 2021	*What Makes a Well-Documented Notebook? A Case Study of Data Scientists' Documentation Practices in Kaggle*	*Computational engineering*	Jupyter	Understand the best documentation practices used by data scientists	80 top-rated notebooks from Kaggle	List of 9 uses for markdown cells (table)	No	No	No
Pimentel et al., 2021	*Understanding and improving the quality and reproducibility of Jupyter notebooks*	*Computational engineering*	Jupyter	Analyse the characteristics of notebooks that limit reproducibility through a series of questions (naming, order of execution, etc.).	1024269 notebooks from GitHub, sub-sample of 38063 popular notebooks; testing of a tool with 12 participants	Details of the structure of notebooks on GitHub and their low reproducibility; List of 8 best practices; popular notebooks more reproducible; many educational notebooks; proposal for a linting tool	No	Best practice and a dedicated tool	No
Chattopadhyay et al., 2020	*What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities*	*Computational Science*	Jupyter, Colab, Databricks, RStudio	Identifying the frictions introduced by the use of notebooks for data scientists	20 interviews and questionnaire (N=156)	Table of identified problems (9)	No	No	Yes, through interviews
Rule et al., 2018	*Exploration and explanation in Computational notebooks*	*Human-Computer interaction*	Jupyter	Describe how data scientists use notebooks	1230000 Notebooks on GitHub; sample of 200 notebooks linked to a scientific publication; 15 interviews with academic data scientists	Structure of GitHub repositories containing notebooks; use of variable text in academic notebooks	Yes, partly	No	Yes, through interviews
Kery et al., 2018	*The story in the notebook: Exploratory data science using a literate programming tool*	*Human-Computer interaction*	Jupyter	Understanding what data scientists retain from their explorations	Interviews with 21 data scientists	Notebooks are often scratch pads and used to share results. There are several strategies for organising the notebook, including "expand and reduce", but generally with a non-linear narrative.	No	Proposal for a historical magnifying glass to go back over the versions	Yes, especially on explorations