Journal of Data Mining and Digital Humanities

EPIsciences overlay journals

Notebook and Open science : toward more FAIR play

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

1 CREM, Université de Lorraine

2 Inno[3]

3 COSTECH – Université de Technologie de Compiègne

4 CREST – GENES/IPP

*Corresponding author: Mariannig Le Béchec mariannig.le-bechec@lorraine.fr

Abstract
Notebooks are now commonly used in digital research practices. Despite their increasing ubiquity, the characteristics, roles, and uses associated with notebooks have seldom been studied from a from a social science perspective. In this article, we present an overview of the available 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 interactive computational notebooks specifically within data science rather than research practices in academic contexts, we discuss the role of notebooks as a vector and lever for the FAIR (Findable, Accessible, Interoperable, Reusable) principles associated with open science.

keywords
notebook, literate programming, jupyter, open science, FAIR

INTRODUCTION

In digital research practices, researchers now 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. The notion of "notebook" refers to largely different items: some are documents that are first written and then compiled to obtain results, such as those made possible by Rmarkdown[1], or directly interactive documents in the browser such as those of the Jupyter[2] project, subsequently referred to as "computational notebooks"[3]. These documents differ from other types of digital

[1] https://rmarkdown.rstudio.com/ (access 03.12.2024)
[2] https://jupyter.org/ (access 03.12.2024)
[3] "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] or "we define a computational notebook as a system that supports literate programming using a text-based programming language (e.g., Python, R, JavaScript, or a DSL) while

documents, which are also referred to as notebooks, such as electronic lab notebooks[4]. These are addressed in the scientific literature[5], and won't be discussed here.
Computational notebooks, which are the outcome of a long history connecting diverse generations of software[6], are now a common tool in teaching computational sciences, supporting data analysis or AI uses. Despite their large adoption and a burgeoning literature on best practices [Davies et al., 2020], there has been a lack of research into their characteristics, roles, and uses in research, especially from the viewpoint of from a social science perspective. In the following article, we wish to provide a comprehensive review of existing empirical studies concerning the use of computational notebooks with a focus on research, delineating current practices, cataloguing the variety of notebook formats, elucidating criticisms, and suggesting enhancements[7]. To do that, we focus on empirical studies aiming to document the evolving practices involving computational notebooks. We searched scientific databases (cf. *Methodology of the literature review*) selecting empirical work. Emphasis was placed on the identification of open science practices and the application of legal principles to notebooks[8].
We observe that the current literature focused primarily on the field of data science and only marginally on research practices themselves. Besides, its main topic is interactive computational notebooks. Existing results showed that the use of computational notebooks led to a diversity of new artefacts with concern regarding their stability and reproducibility. Consequently, we discuss the role of computational notebooks in promoting the FAIR (Findable, Accessible, Interoperable, Reusable) principles of open science. We call for more dedicated study on how research practices and ecosystems evolve with the rise of this new media.

METHODOLOGY OF THE LITERATURE REVIEW

The diffusion of computational notebooks led to a diversity of analysis, ranging from personal feedback to dedicated academic studies. Facing this emerging literature not yet stable at the interface between computer sciences, human, and computer relation and social sciences, our primary aim was to establish the current state of knowledge on notebooks practices (2023). To achieve this, we conducted a non-systematic 'scoping review' on existing published results.
Our research concentrated on empirical work analyzing current practices with computational notebooks. We queried several databases, including Google Scholar, Scopus, Dimensions, BASE Api, and Semantic Scholar, as well as the Zotero collection of the Notebook Working Group[9]. We used the following keywords and combinations: '*notebook*' OR '*interactive*

---

interweaving expository text and program outputs into a single document." [Lau et al., 2020]. Adding to that, notebooks can be interactive, thus a distinction can be drawn between computational notebooks that are interactive and those that are non-interactive. In the text, we use mainly computational notebooks to name them, shorten as notebook for simplicity.

[4] The electronic lab notebook is the dematerialised version of the lab notebook, which has been used for ages to track the daily scientific activities (experiment description, protocols used, etc.) The lab notebook has also a legal value (proof of an invention or a discovery).

[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 (access 03.12.2024)

[6] The history of computational notebooks, since the development of interactive interfaces such as Matlab or Mathematica in the 80's, is still to be told extensively, but it connects changes in scientific programming, the evolution of computational tools such as Maple and the rise of interactivity in browsers [Schultz, 2023].

[7] We would like to thank the three reviewers who helped to improve this article (Nicolas Sauret, Arthur Perret and Nicolas Thiery), and the journal editors (Marie-Laure Massot and Julien Cavero) for their support. We added some of their ideas in the article.

[8] This preliminary work outlines the methodology of NOOS (Notebook for Open Science) project, funded by the GIS Réseau URFIST 2022. It clarifies the issues that will be addressed in this research project.

[9] https://www.zotero.org/groups/4416056/gt-notebooks/items/PGT9T72T/library (access 03.12.2024)

notebook' OR '*computational notebook*' OR '*executable notebook*' OR '*jupyter notebook*'. We didn't limit our search to open science keywords to expand the scope.
Then, the four authors conducted a systematic reading of the articles to filter contributions dedicated to investigate practices with computational notebooks. Contributions lacking investigation, limited to personal viewpoint, or only discussing software features were excluded[10]. For instance, we chose to discard [Lau, 2020] that deals with computational notebooks as software systems. This led us to select 12 articles we then compared regarding their field, scope, methodologies, and results. After reading the corpus, we identified emergent topics we analyze in the results (cf. *Appendix table*).

I RESULTS

1.1 An emerging literature focusing on software engineering and data science

The literature dedicated to notebooks is recent and addresses primarily computer and software engineering, as well as human-software interaction. IT engineers, data scientists, and academics alike use these software environments developed by a diversity of contributors from both academia and private sector. The distinction between industrial and scientific purposes can be blurred, making it challenging to summarise practices. The studies often look to find solutions to problems identified in the context of application. 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 highlight solutions that are designated or refer to already existing applications [Chattopadhyay et al., 2020; Kery et al., 2018].
Issues related to academic research are sometimes addressed, especially when contributors originate from an academic institution [Quaranta et al., 2022]. However, in the case of interviews with academic actors, there is no discussion on field specificities [Quaranta et al., 2022; Rule et al., 2018]. In the literature on scientific programming, a well-identified tension between researchers and software engineers is 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 addresses a few specific issues, such as the dissemination of results. If a few articles focus on scientific research, they don't tackle the specificities of academic activities [Samuel & Mietchen, 2022; Wagemann et al., 2022].
Existing articles focus 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]. Additionally, the articles do not address the factors that contribute to the spread of practices or the limitations of adoption. For instance, there is no data on the percentage of users based on their professional field or discipline, or on alternative applications. 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 are on analysing digital traces of notebooks. These notebooks are extracted from GitHub, a collaborative code version

[10] Since this topic is crossing a diversity of fields mainly outside academia, this criteria allowed us to decide how to include a study in our analysis.

management platform, and Kaggle[11], a data science machine learning competition platform, both widely used in the IT industry. This approach allows 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[12] [Raghunandan et al., 2023]. The availability of digital data thus enables the definition of 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, there are limits in the representativeness of those samples. The authors investigate both the contexts in which notebooks are created and the users themselves to a limited extent, relying 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 also include semi-structured interviews with notebook users. While these interviews investigate practices and judgements made about notebooks, their focus remains confined to the notebook itself, with little interest for the broader work context and associated activities in which the notebook finds application. For instance, dimensions such as the socialisation and training with these tools were not considered in these interviews. Only one study mentions administering a questionnaire [Chattopadhyay et al., 2020]. In research aimed at prototyping new tools, UX test methods are 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.

1.2 The Rise of Jupyter Notebooks

Overall, it appears that the literature primarily focuses on computational notebooks, particularly those from the Jupyter project. While other computational notebook solutions exist [Lau et al., 2020], for instance R-markdown/R-Studio (now Posit[13]), SageMath[14] or Observable[15], or multiple variations of online services with partial compatibility (Google Colab, Microsoft Azure, Kaggle, Databricks, Apache Zeppelin), they are less investigated.

Although several studies [Pimentel et al., 2021; Samuel & Mietchen, 2022] highlight the importance of reproducibility, the literature put the emphasis on the lack of thereof, especially on GitHub (cf. 1.4 *Identification of numerous limitations and criticisms*). However, this criticism is not specific to computational notebooks and applies to code [Trisovic et al., 2022][16]. Researchers identify various 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]. Other factors, such as notebook popularity, could also have an influence. Traditionally, reproducibility refers 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 integration.

---

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

[12] Save changes in the source repository: https://git-scm.com/docs/git-commit/fr (access 03.12.2024)

[13] https://posit.co (access 03.12.2024)

[14] https://www.sagemath.org/ (access 03.12.2024)

[15] https://observablehq.com/ (access 03.12.2024)

[16] [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.”

ISSN 2416-5999, an open-access journal

Studies suggest that notebooks, as computing practices, 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 programming practices, notebooks offer greater freedom and a constant tension between quality and speed of development [Quaranta et al., 2022]. They seem to be characterised by a specific programming style [Grotov et al., 2022] and introduce new ways of documenting code [Wang et al., 2021] The potential decline in quality[17] may arise from the use of notebooks by multi-disciplinary teams, presenting challenges to ensuring consistent programming practices and documentation [Wang et al., 2021].

Studies of large samples of notebooks, particularly those available on GitHub [Rule et al., 2018], show their diversity and open the way to classifications. Notebooks have diverse uses, such as teaching or demonstrating materials, as a step in data analysis, or as a training material. The categorization of [Liu et al. 2021] proposes five types of notebooks:

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

Pedagogical uses appear 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 non-linear and evolving nature of computational notebooks is widely emphasised [Ramasamy et al., 2023]. A constant observation is a bipolarity of practices: exploratory vs. explanatory. 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 focuses 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.

1.3 Studying notebooks to recommend best practices

The literature frequently addresses the question of 'best practices', which can refer to effective methods or quality standards. Observations and recommendations frequently overlap across studies, indicating that a comprehensive coverage is achieved: ‘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].

[17] Another explanation is suggested by Nicolas Thiéry, in the review discussion of this article, that there is a distinction to make between programming and computing. Computation focuses on obtaining specific results, while programming is oriented toward developing reusable software solutions. Interactive computational notebooks are primarily used for computation rather than traditional programming. Consequently, programming practices such as modularity, reusability, documentation, testing, and maintenance—while essential in software development—are often relaxed or applied less rigorously in the context of computation.

ISSN 2416-5999, an open-access journal

The general observation is that these ‘best practices’ were known, but mainly not applied [Quaranta et al., 2022]. [Pimentel et al., 2021] provide eight recommendations for notebook reproducibility, which can be summarised in a table (Table 1).

General categories	Best practices
Make your analysis traceable and reproducible	• Use a version control system to manage project dependency • Manage project dependencies • Provide 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
Exploit the paradigm of literate programming	• Document your code for yourself and others • Use Markdown headings to structure your notebook
Keep your notebook clear and concise	• Keep your notebook clear • Keep your notebook concise
Differentiate between artefacts produced during development and production	• Differentiate between artefacts produced during development and production
Adopt open distribution	• Make your notebook available • Make 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 is also the matter of the integration of the notebook into a wider environment. Indeed, notebooks rely on a broader software environment, either during their inception or reuse phase. 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 are not addressed.

1.4 Identification of numerous limitations and criticisms

Notebooks are often praised for their educational value, whether for presenting courses [Wagemann et al., 2022] or exercises and tutorials [Pimentel et al., 2021]. However, they are also subject to numerous criticisms, from downloading data to disseminating or collaborating [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 notebook. In response to criticisms, researchers often made proposals for improving or remedying the shortcomings of notebooks [Quaranta et al., 2022; Ramasamy et al., 2023; Rule et al., 2018].
Academic research practices appear to be often hindered by poor reproducibility, as studies analysing many notebooks have evidenced. The studies trying to reproduce notebooks obtains a really low reproducibility rate, with the first issue not being able to run the notebook (22 to 26% can be run) and then to produce the same results (around 5 to 15%). If all notebooks available don't originate from research, and are not built to be reproducible, this problem of reproducibility is also identified within academic production [Samuel & Mietchen, 2022].

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 are used for exploration rather than to explain and narrate results, several essential pieces of documentation are 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 [Wang et al., 2021]. Regarding reproducibility and reuse, it is 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, fueling an ongoing search for better tools. A UX questionnaire proposed by Jupyter highlights 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 is frequently low [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 is suspected to be connected to the very use of the notebook as an exploratory tool rather than for demonstrating or presenting results. Speed is favoured to quality, and this is close to the idea of the 'messy notebook' that emerged in many articles.

Furthermore, the notebook format receives criticism due to the constraints associated with its interface, which can result in errors or lower quality code. For instance, some articles note 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 non-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 are proposed to address these areas of friction.

These solutions include facilitating the security of confidential data and preserving modification histories, both of which are considered major challenges.
Overall, the literature reviews the challenges faced by notebooks for reproducibility. In the following section, we broaden the scope and explore further challenges faced by notebooks for open science (availability of data, conditions for re-use, use of licences).

II DISCUSSION: ARE NOTEBOOKS A STRATEGIC TOOL FOR OPEN SCIENCE?

Building on this literature review, we want to focus on the relationship between notebooks and open science. Open science aims to promote the dissemination of knowledge as soon as it is available, using digital and collaborative technologies[18]. Articles, data, and scripts (source codes) are the main elements targeted for sharing and openness. Open access 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. This even though the open source and free software movements have been at the forefront of the grassroot open science movement for more than four decades.

We propose three main goals in order to clarify the relation between computational notebooks practices and science perspectives:

- a better understanding of the practices of research professionals;
- an explicit discussion of the issues surrounding the FAIRisation of notebooks;
- the contextualisation of the notebooks in the wider environment of free circulation of scientific codes and productions, in particular through open licences.

2.1 What are research professionals doing with notebooks?

Existing studies focus 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 lack of specific knowledge on those fields. For instance, there is a need to better delineate the differences between disciplines, researchers positions (status and age/generation), and their methods. According to several studies, there is indeed variability in the digital practices of research professionals[19]. Although there are testimonials and feedback [Beg et al., 2021], there is little work dedicated to the practices of research professionals in relation to computational notebooks: the only systematic article deals with the biomedical field [Samuel & Mietchen, 2022].
More specifically, computational notebooks are often presented as a new interface between source code and publication via executable articles. For example, 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*[20]. 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

[18] 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 (access 03.12.2024)
[19] For example, « Pratiques et usages des outils numériques dans les communautés scientifiques en France » [Le Béchec et al., 2022] or the study « Décliner la Science Ouverte » [Gruson-Daniel & Groupe Projet Réussir L'Appropriation De La Science Ouverte, 2022].
[20] https://www.theatlantic.com/science/archive/2018/04/the-scientific-paper-is-obsolete/556676/ (access 03.12.2024)

ISSN 2416-5999, an open-access journal

media (cf. 2.3 *Contextualising notebooks within their 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[21], 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 does not address the conditions that lead a user to start using a notebook or the interdependence between notebooks and other tools such as programming scripts, software (such as IDE), 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 are addressed (cf. 1.4 *Identification of numerous limitations and criticisms*), highlighting the 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.

2.2 Moving towards the FAIRisation of notebooks

Despite the fact that the genesis of projects, such as Jupyter, is directly in line with the open science perspective, the relationship between notebooks and open science, in terms of best practices, are rarely mentioned. Most research on notebooks and their improvements focus on data engineering applications. Although recommendations on formatting, writing, and tools for improving practices often accompany criticisms directed at notebooks, attention is rarely given to the dimensions of dissemination and openness. In the articles examined, advice on dissemination remains general (Table 1). The FAIR principles, which incentised for example the use of persistent identifiers such as the DOI, are not mentioned. Currently, there is a lack of shared guidelines to facilitate the application open science principles and FAIR data management in the notebook environment[22]. Different solutions have been suggested to improve identified limitations, but there is still little evidence for their adoption[23].

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

[21] For instance, there is a book dedicated on how to use Jupyter tools in a pedagogical context https://jupyter4edu.github.io/jupyter-edu-book/ (access 03.12.2024)
[22] 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]. There are field-specific initiatives, such as *Notebooks Now* in geophysics, that propose new workflows centered around notebooks: https://data.agu.org/notebooks-now/about.html (access 03.12.2024)
[23] The scientific community provides tooling to improve notebooks as FAIR scientific objects. For example MyST helps to provide metadata with Front Matter https://mystmd.org/guide/frontmatter (findability) (access 03.12.2024). Other tools like nbdime https://nbdime.readthedocs.io/en/latest/ (access 03.12.2024) and text-based notebooks help with version control. Nbval https://nbval.readthedocs.io/en/latest/index.html (access 03.12.2024) provides help for testing. Moreover, tools like binder and package managers are crucial to build reproducible software environments. Nbviewer https://jupyter-tutorial.readthedocs.io/en/24.1.0/nbviewer.html (access 03.12.2024) or the rendering of notebooks in software forges help with accessibility. But most of these tools are not included in our literature review.

literate 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 and learn computational data analysis in various scientific fields [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 more general challenges of open science and reproducible computing. As with other digital practices, several areas require further exploration, such as ensuring complete references to sources and data, standardizing 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.

2.3 Contextualising notebooks within their 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 shows that existing work focuses almost exclusively on notebooks themselves (uses or content). However, the extent of their integration into digital ecosystems is little 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 licence on Jupyter notebooks, with at least a third of resources lacking licences [Schröder et al., 2019]. Even if this absence of licence exists also for other pieces of code, the use of notebook by a broader audience, not familiar with programming management, may increase the lack of familiarity for their creator. When licences are present, the most common ones are MIT, GPLv and CC0. However, the authors do 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.

Conclusion

The scientific literature on the computational notebooks, which focus mainly on Jupyter, reveals new practices and opens the discussion on how to stabilise them. Although the genesis of the Jupyter (and its predecessor Ipython) that contributed to the wide adoption of this interface is intrinsically linked to the issue of open source and the openness of scientific knowledge [Schultz, 2023], 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 review of dedicated empirical studies on computational notebook practices. It described 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. If 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.). In fine, it is the notebooks' integration into a broader research infrastructure, tools and open standard ecosystem that is key to promote the open circulation and exploitation of knowledge.

This review showed that the use of computational notebooks in and out academic work is still an emergent field, and call for more studies. Indeed, almost all the points discussed are not specific to notebooks and can be found in other programming practices. However, they are amplified by the multiplication of users brought by these new tools. Consequently, there is for instance a need to conduct dedicated studies on how research infrastructures (at local, national, and European levels) have implemented notebook platforms for their communities. There is also a need to better understand the diversity and the history of notebook systems to explain their diffusion across various communities. As part of the NOOS project continuation, we will focus on the next steps to better understand 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. It would be beneficial to extend the analysis of the relationship between science and technology in a political, economic and social context that is favourable to open science but sometimes detached from the day-to-day realities of research.

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/

Davies, A., et al., "Using Interactive Digital Notebooks for Bioscience and Informatics Education," *PLoS Computational Biology* 16, no. 11 (2020): 1–19, doi:10.1371/journal.pcbi.1008326.

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

ISSN 2416-5999, an open-access journal

Gruson-Daniel, C., Groupe Projet Réussir L'Appropriation De La Science Ouverte (2022). Décliner la science ouverte. [Rapport de recherche] Comité pour la science ouverte. 149 p. https://dx.doi.org/10.52949/27

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

Le Béchec, M., Bouchard, A., Charrier P., Denecker C., Gallezot, G., Rennes, S. (2022). Pratiques et usages des outils numériques dans les communautés scientifiques en France. [Rapport de recherche] Comité pour la science ouverte. 2022, 112 p. https://dx.doi.org/10.52949/5

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

Lau, S., Drosos, I., Markel, J., Guo, P. (2020). « The Design Space of Computational Notebooks: An Analysis of 60 Systems in Academia and Industry ». *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC* 2020-Augus. doi:10.1109/VL/HCC50065.2020.9127201.

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

Appendix: table of the 12 identified references

Reference	Title	Speciality	Type	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	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

Reference	Title	Speciality	Type	Objective	Method	Main finding	Science	Recommendations	Practices
Wagemann et al., 2022	*Five Guiding Principles to Make Jupyter Notebooks Fit for Earth Observation Data Education*	Earth 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 notebooks	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	Numerous interview transcripts
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

Reference	Title	Speciality	Type	Objective	Method	Main finding	Science	Recommendations	Practices
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
Kerry 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