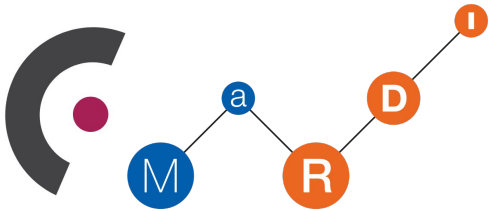
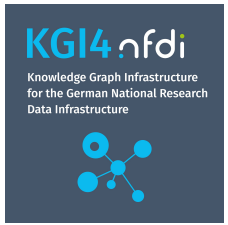
Analyzing the reproducibility of research-related Jupyter notebooks at scale

Daniel Mietchen (FIZ Karlsruhe & MaRDI & KGI4NFDI) & Sheeba Samuel (TU Chemnitz)





TECHNISCHE UNIVERSITÄT
CHEMNITZ

SE25, 26 February 2025 -
DOI: 10.5281/zenodo.14678155 (slides) / 10.18420/se2025-07 (proceedings)

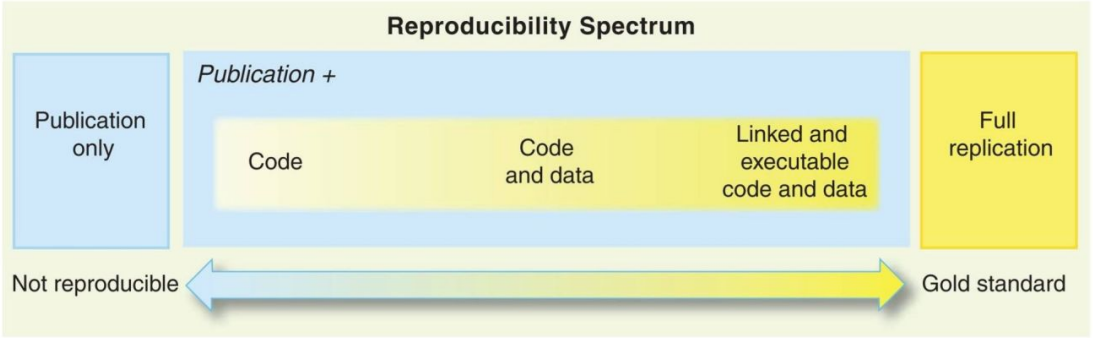
Outline

- What did we do?
- Why did we do this?
- How did we do this?
- Results
- What’s next?
- Discussion

What did we do?

- We studied
  - the computational reproducibility
    - of 27,271 Jupyter notebooks
      - from 2,660 public GitHub repositories
        - associated with 3,467 peer-reviewed biomedical publications
          - from 740 journals

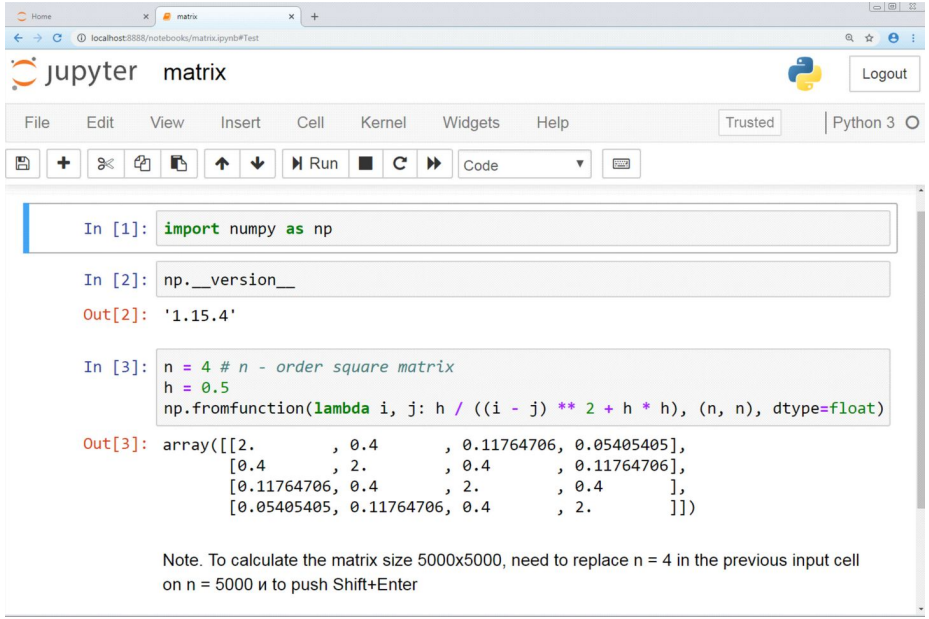
- Key concepts here
  - computational reproducibility
  - Jupyter notebook
  - GitHub repository
  - peer-reviewed publication
  - PubMed Central
    - full-text XML of biomedical papers
    - https://pmc.ncbi.nlm.nih.gov/



[Reproducible Research in Computational Science, Roger D. Peng, 2011]

What are Jupyter notebooks?

- They
  - provide a shared environment for
    - code
    - code execution
    - code comments
    - Markdown-based documentation
  - run on free and open-source software
  - can be shared conveniently
  - have been the subject of guidelines
  - can assist in enhancing computational reproducibility
    - prior studies found problems for generic Jupyter notebooks
    - perhaps this works better for Jupyter notebooks associated with peer-reviewed papers?



Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks

Adam Rule[1], Amanda Birmingham[2], Cristal Zuniga[3], Ilkay Altintas[4], Shih-Cheng Huang[4¤], Rob Knight[3,5], Niema Moshiri[6], Mai H. Nguyen[4], Sara Brin Rosenthal[2], Fernando Pérez[7], Peter W. Rose[4*]

A Large-scale Study about Quality and Reproducibility of Jupyter Notebooks

João Felipe Pimentel*, Leonardo Murta*, Vanessa Braganholo*, and Juliana Freire†

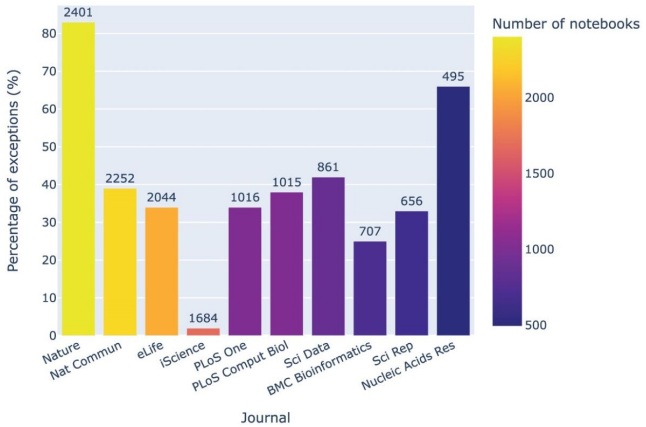
Why did we do this?

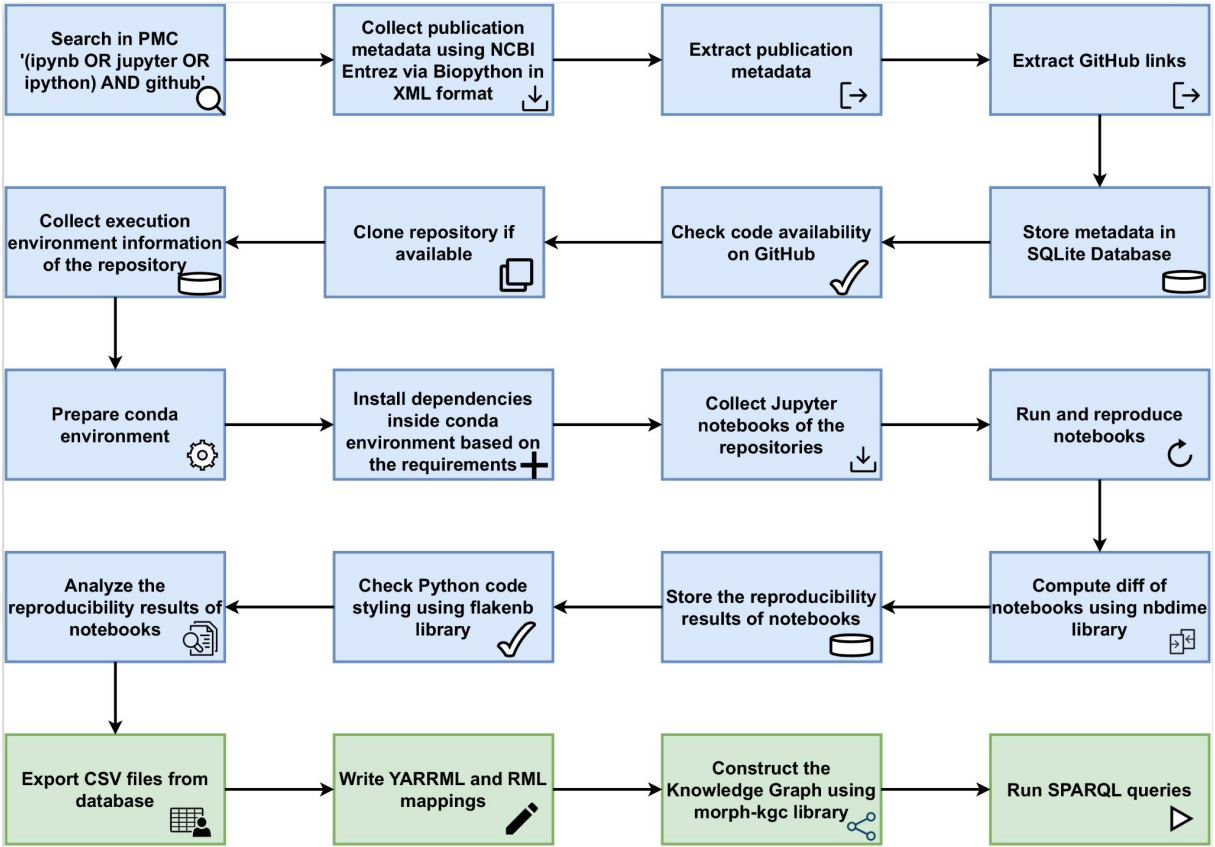
- We wanted to find out
  - How does the computational reproducibility of Jupyter notebooks work in practice?
    - What are common issues?
    - What are best practices?
    - How does computational reproducibility vary
      - by research field
      - by journal
      - over time
      - …
  - because insights into these questions might affect research and education around
    - computational reproducibility
    - scientific reproducibility
    - (research) software engineering
    - software documentation
    - and related matters

How did we do it?

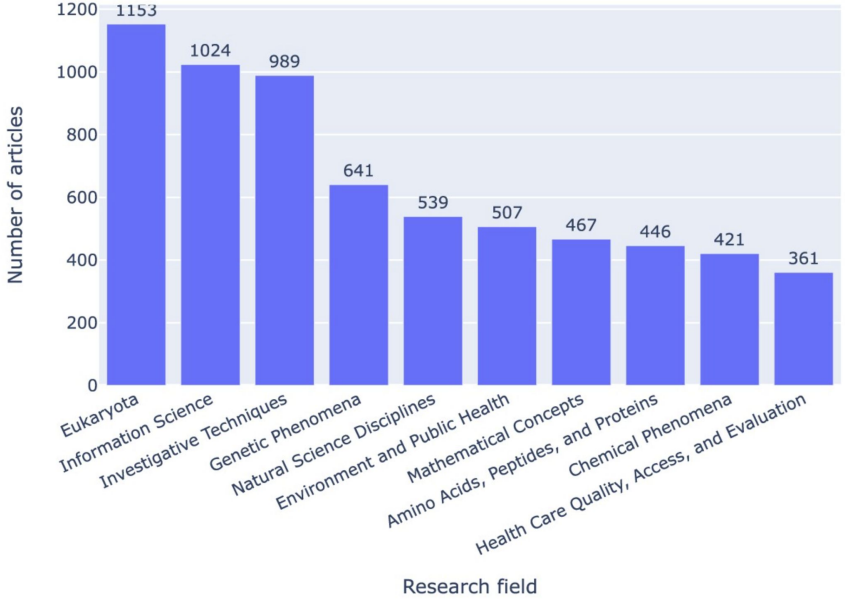
740	3,467	2,660	27,271
Journals	Publications	Repositories	Notebooks

43 days

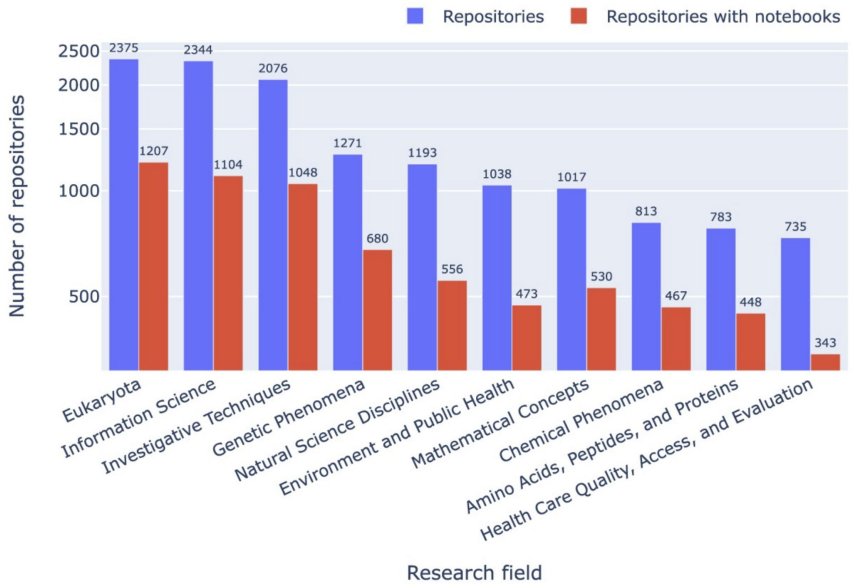




Results - Research Field



SPARQL Query to reproduce the figure: [Research articles by research field]



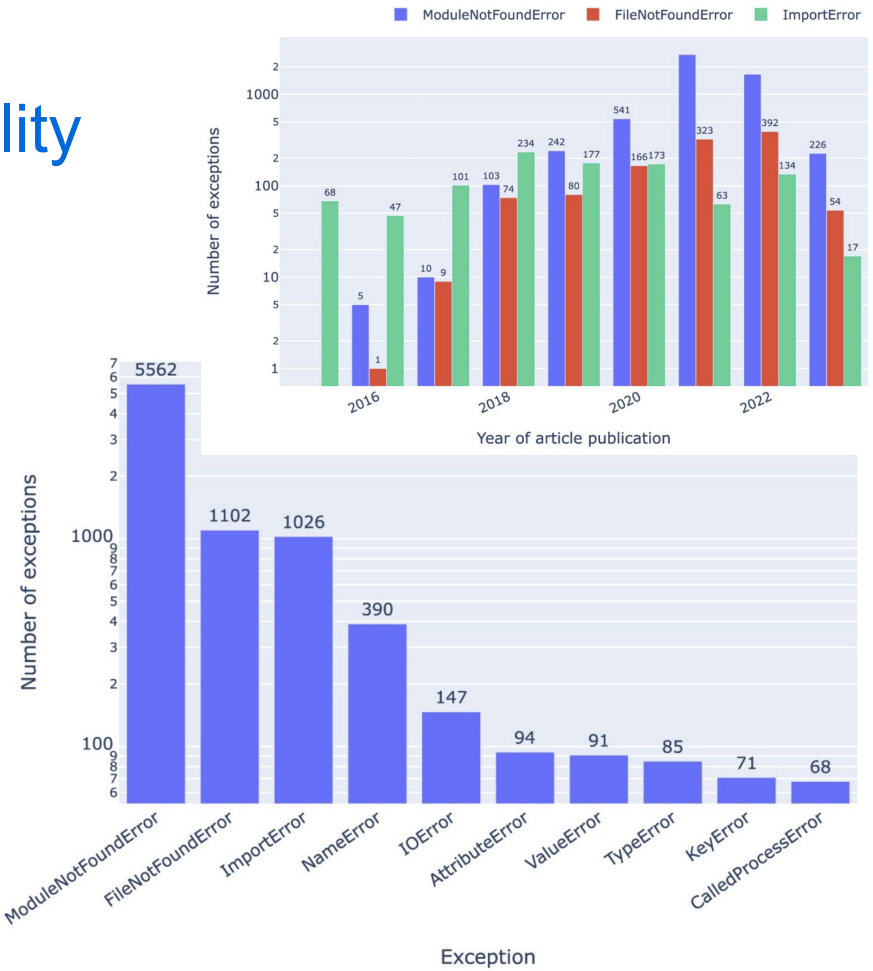
SPARQL Query: [Research field (MeSH terms) by the number of GitHub repositories that contain at least one Jupyter notebook.]

Results - Programming Languages of Jupyter notebooks

Notebook language	Rerun	Initial run	Ratio rerun/initial
Matlab	134	9	14.9
Julia	295	59	5.0
Unknown	3,112	720	4.3
Python	22,578	8,160	2.8
R	891	461	1.9
Bash	36	24	1.5
Sos	29	24	1.2
Groovy	95	95	1.0
Scala	29	29	1.0
Java	8	8	1.0

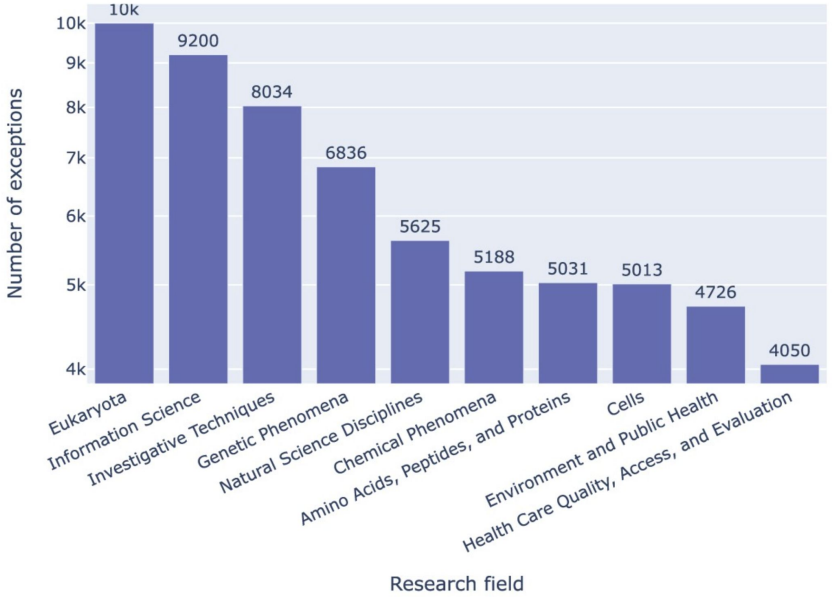
Results - Notebook Reproducibility

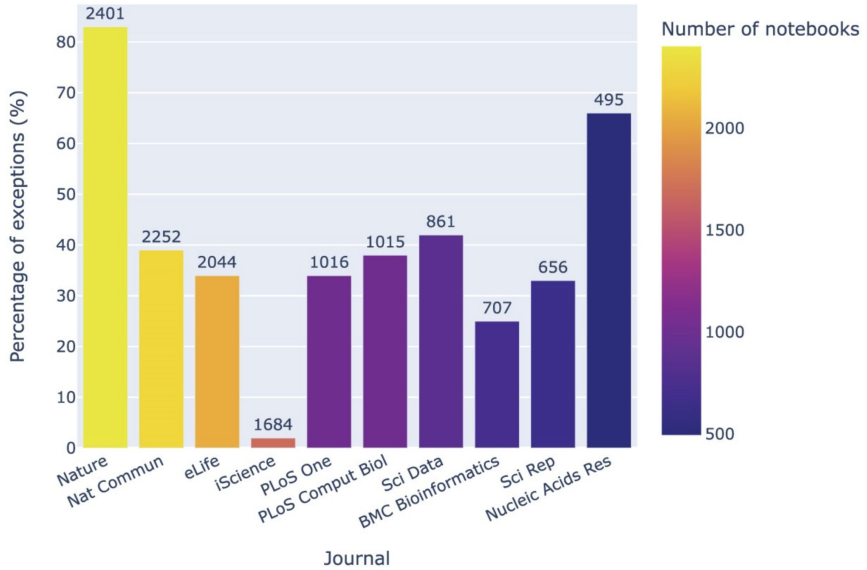
Features	Notebooks with `different` results	Notebooks with `identical` results
Number of notebooks	324 (⊠ 151)	879 (⊠ 245)
setup.py	0 (⊠ 0)	344 (⊠ 98)
requirement.txt	1 (⊠ 0)	353 (⊠ 107)
pipfile	0 (⊠ 0)	0 (⊠ 0)
Total cells	23 (⊠ 17.9)	19.6 (⊠ 17.1)
Code cells	15.4 (⊠ 12.3)	11.3 (⊠ 9.8)
Markdown cells	7.6 (⊠ 5.6)	8.3 (⊠ 6.7)
Ratio of Markdown vs. code cells	0.49 (⊠ 0.46)	0.73 (⊠ 0.68)
Empty cells	0.9 (⊠ 0.7)	0.7 (⊠ 0.7)
Differences	6.3 (⊠ 5.3)	0 (⊠ 0)
Execution time (s)	18.3 (⊠ 22.1)	57.6 (⊠ 16.4)
Execution time per code cell (s)	1.88 (⊠ 1.80)	5.09 (⊠ 1.67)



SPARQL Query: Exceptions occurring in Jupyter notebooks in our corpus.

Results - Notebook Reproducibility





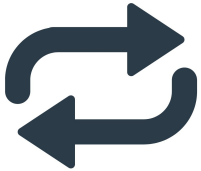
SPARQL Query: [Jupyter notebook exceptions by research field, taking as a proxy the highest-level MeSH terms of the article associated with the notebook.]

Effect of peer review and editorial workflows: Nature has about 50 *percent* more notebooks than iScience, but Nature's fail about 40 *times* more often

Environmental Footprint

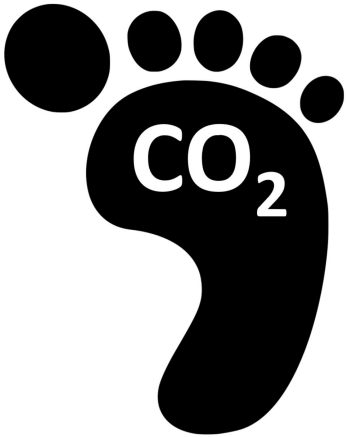
Simplified calculation: carbon footprint = energy needed * carbon intensity

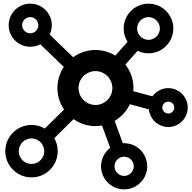
(http://calculator.green-algorithms.org/)



373.78 kWh

126.58 kg CO2e





65.97 Wh

7.33 g CO2e

Limitations

- only computational reproducibility
- only for Jupyter notebooks from articles available through PubMed Central
- only ran notebooks written in Python
- fully automated pipeline not using LLMs, ignoring non-FAIR instructions
- stops at first error, does not attempt fixes
- ecological footprint only assessed for running the pipeline, not developing it

Further details

- Analyzing the reproducibility of research-related Jupyter notebooks at scale
- Initial Run
  - 2021
  - Preprint: https://doi.org/10.48550/arXiv.2209.04308
- Rerun
  - 2023
  - Code : https://github.com/fusion-jena/computational-reproducibility-pmc
  - Data : https://doi.org/10.5281/zenodo.8226725
  - Paper : https://doi.org/10.1093/gigascience/giad113
- Knowledge graph (see also deRSE25 talk today at 16:00)
  - Website: https://w3id.org/fairjupyter
  - SPARQL Endpoint: https://reproduceme.uni-jena.de/#/dataset/fairjupyter/query
  - Knowledge Graph Browser: https://reproduceme.uni-jena.de/fj/
  - Sample query: PLOS Comp Bio articles and associated GitHub repos and Jupyter notebooks
  - Preprint: https://doi.org/10.48550/arXiv.2404.12935
  - Paper: https://doi.org/10.4230/TGDK.2.2.4

What's next?

- Assessments of Jupyter notebooks in single GitHub repos
  - focus here is on evaluations before submission to journals/ conferences
- Assessments of dockerized Jupyter notebooks
- Assessments of non-Python notebooks
- Assessments of Jupyter notebooks from publications not in PubMed Central
- Software/ Data Carpentry course based on FAIRJupyter
- FAIRJupyter as a continuous service
- Your ideas here - collaborations welcome

Thank you

- German Research Foundation (DFG), TRR-386, project number 514664767
- Alfred P. Sloan Foundation under grant number G-2021-17106.
- Ara Cluster of Friedrich Schiller University Jena under DFG grants INST 275/334-1 FUGG and INST 275/363-1 FUGG.
- BASE4NFDI/KGI4NFDI: DFG 521453681

- **Contact**: sheeba.samuel@informatik.tu-chemnitz.de & daniel.mietchen@fiz-karlsruhe.de

SE25, 26 February 2025

DOI: 10.5281/zenodo.14678155 (slides) / 10.18420/se2025-07 (proceedings)